



**TUGAS AKHIR - TE 145561**

## **SISTEM OTOMASI PENYEMAIAN BENIH SAYURAN HIDRO-PONIK PADA KEBUN SAYUR SURABAYA**

Moh. Ilham Aziz W.  
NRP 2214030030  
Safitri Febrianti  
NRP 2214030046

Dosen Pembimbing  
Rachmad Setiawan, ST., MT.  
Suwito, ST., MT.

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**FINAL PROJECT - TE 145561**

***AUTOMATION SYSTEM OF HYDROPONIC SEED VEGETABLES IN KEBUN SAYUR SURABAYA***

Moh. Ilham Aziz W.  
NRP 2214030030  
Safitri Febrianti  
NRP 2214030046

Supervisor  
Rachmad Setiawan, ST., MT.  
Suwito, ST., MT.

COMPUTER CONTROL STUDY PROGRAM  
Electrical and Automation Engineering Department  
Vocational Faculty  
Sepuluh Nopember Institute of Technology  
Surabaya 2017




## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Sistem Otomasi Penyemaian Benih Sayuran Hidroponik pada Kebun Sayur Surabaya**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Moh. Dham Aziz W.  
NRP 2214030030

Surabaya, 17 Juli 2017



Safitri Febrianti  
NRP 2214030046

-----Halaman ini sengaja dikosongkan-----

# **SISTEM OTOMASI PENYEMAIAN BENIH SAYURAN HIDRO- PONIK PADA KEBUN SAYUR SURABAYA**

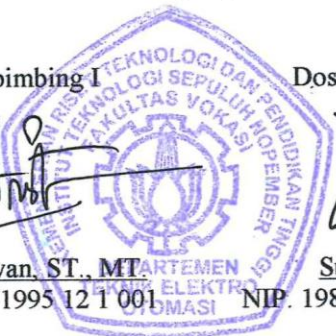
## **TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada  
Program Studi Komputer Kontrol  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II



Rachmad Setiawan, ST., MT. Suwito, ST., MT.  
NIP. 1969 05 29 1995 12 1 001 NIP. 1981 01 05 2005 01 1 004

**SURABAYA  
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----



# **SISTEM OTOMASI PENYEMAIAN BENIH SAYURAN HIDROPONIK PADA KEBUN SAYUR SURABAYA**

**Nama** : Moh. Ilham Aziz W.  
Safitri Febrianti  
**Pembimbing** : Rachmad Setiawan, S.T., M.T.  
Suwito, S.T., M.T.

## **ABSTRAK**

Tahap penyemaian benih merupakan salah satu tahap yang penting pada pertumbuhan tanaman hidroponik karena tahap ini menyangkut kelangsungan pertumbuhan tanaman pada fase selanjutnya. Beberapa faktor yang dapat memerangui pertumbuhan tanaman diantaranya temperatur, kelembaban, intensitas cahaya, dan kadar air. Disamping itu, selama ini pemilik kebun sayur hidroponik masih melakukan pengontrolan keempat parameter tersebut secara manual. Hal ini dirasa kurang efektif dan efisien, karena pemilik kebun sayur harus melakukan pengontrolan secara langsung di kebun sayurnya.

Berdasarkan kondisi ini, dibuat suatu alat yang dapat melakukan pengontrolan pada tahap penyemaian tanaman hidroponik. Metode yang digunakan pada sistem alat ini yaitu dengan melakukan pengontrolan keempat parameter. Mikrokontroler yang digunakan adalah Arduino Mega 2560 yang telah diterapkan program inisialisasi dan konfigurasi perangkat keras serta pembacaan sinyal masukan dari sensor temperatur dan kelembaban HSM-20G, intensitas cahaya pada LDR, dan ketinggian air pada *Water Level Funduino*. Selain itu, terdapat aplikasi *android* yang berfungsi untuk memonitoring dan mengontrol aktuatur pada alat yang terintegrasi via *bluetooth*.

Hasil dari penelitian ini adalah sistem dapat menjaga temperatur, kelembaban dengan presentase *error* 0-0,7% yakni pada temperatur 27-29°C, ketinggian air dengan presentase *error* 0,4-1,3% dan didapatkan presentase keberhasilan alat 88,89 %, serta aplikasi yang dibuat telah dapat digunakan untuk memonitoring dan melakukan pengontrolan kondisi tanaman secara *real time*.

**Kata Kunci** : Hidroponik, *Humidifier*, Sensor HSM-20G, *Water Level Funduino*, *bluetooth*, *Android*.

-----Halaman ini sengaja dikosongkan-----

## ***AUTOMATION SYSTEM OF HYDROPONIC SEED VEGETABLES IN KEBUN SAYUR SURABAYA***

**Name : Moh. Ilham Aziz W.  
Safitri Febrianti**  
**Supervisor: Rachmad Setiawan, S.T., M.T.  
Suwito, S.T., M.T.**

### ***ABSTRACT***

*Seeds seeding stage is one of the important stages in the growth of hydroponics plants because this stage concerns the continuity of plant growth in the next phase. Some factors that can affect the growth of plants such as temperature, humidity, light intensity, and moisture content. In addition, so far the owner of hydroponic vegetable garden still controls the four parameters manually. This is considered less effective and efficient, because the owner of the vegetable garden should do the control directly in the vegetable garden.*

*Based on this condition, a tool is made to control the hydroponic plant seeding stage. The method used in this tool system is by doing the control of the four parameters. Microcontroller used is Arduino Mega 2560 which has implemented initialization program and hardware configuration and input signal readings from HSM-20G temperature and humidity sensors, light intensity at LDR, and water level at Water Level Funduino. Beside that, there is an application on android smartphone to monitor and control actuators that integrated with bluetooth.*

*The result of this research is the system can keep maintain temperature, humidity with error percentage 0-0.7% at 27-29°C, water height with percentage error 0.4-1.3% and got percentage of success 88.89% of system, also the application that can be used to monitor and controlling crop conditions in real time.*

***Keywords : Hydroponics, Humidifiers, HSM-20G Sensors, Funduino Water Level, bluetooth, Android.***

-----Halaman ini sengaja dikosongkan-----

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

### **SISTEM OTOMASI PENYEMAIAN BENIH SAYURAN HIDROPONIK PADA KEBUN SAYUR SURABAYA**

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Rachmad Setiawan, ST., MT. dan Bapak Suwito, ST., MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 17 Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
ABSTRAK .....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xix
DAFTAR TABEL.....	xxiii
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	1
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Metodologi Penelitian .....	3
1.6 Sistematika Laporan.....	3
1.7 Relevansi.....	4
<b>BAB II TEORI DASAR.....</b>	<b>7</b>
2.1 Hidroponik .....	7
2.1.1 Teori Dasar Hidroponik .....	7
2.1.2 Hidroponik Sistem <i>Wick</i> .....	8
2.1.3 Sayuran Hidroponik .....	8
2.1.4 Selada ( <i>Lactuca sativa L.</i> ) .....	9
2.2 Teori Komunikasi Data.....	9
2.3 Arduino Mega 2560 .....	10
2.4 Arduino IDE.....	12
2.5 TRIAC <i>Switching</i> .....	13
2.6 <i>Power Supply</i> .....	14
2.7 Sensor HSM 20-G .....	17
2.8 Sensor Ketinggian Air.....	18
2.9 Sensor Cahaya LDR.....	20
2.10 RTC ( <i>Real Time Clock</i> ).....	21
2.11 <i>Driver Motor</i> .....	22

2.12	<i>Relay</i> .....	23
2.13	<i>Ultrasonic Humidifier</i> .....	23
2.14	Modul <i>SD Card</i> .....	24
2.15	<i>Liquid Cristal Display</i> (LCD) 20x4.....	25
2.16	Pompa Air.....	26
2.17	<i>Bluetooth</i> .....	27
2.18	Sistem Operasi <i>Android</i> .....	30
2.19	App Inventor.....	31
<b>BAB III PERANCANGAN <i>HARDWARE</i> DAN <i>SOFTWARE</i> .....</b>		<b>33</b>
3.1	Perancangan <i>Hardware</i> .....	34
3.1.1	Perancangan Rangkaian <i>Shield</i> Arduino Mega 2560... ..	34
3.1.2	Perancangan Rangkaian <i>Solid State Relay</i> (SSR) Menggunakan TRIAC .....	37
3.1.3	Perancangan Rangkaian <i>Power Supply</i> .....	39
3.1.4	Perancangan Rangkaian Pengondisi Sinyal Sensor HSM 20-G.....	40
3.1.5	Perancangan Sensor <i>Water Level Funduino</i> .....	40
3.1.6	Perancangan Sensor Cahaya LDR .....	41
3.1.7	Perancangan RTC( <i>Real Time Clock</i> ) DS1307.....	42
3.1.8	Perancangan <i>Driver</i> Motor L298N.....	43
3.1.9	Perancangan <i>Hardware Plant</i> .....	46
3.1.10	Perancangan <i>Relay</i> sebagai <i>Switching</i> .....	46
3.1.11	Perancangan Modul <i>SD Card</i> .....	47
3.1.12	Pengaturan Modul <i>Bluetooth</i> HC-05 .....	47
3.2	Perancangan <i>Software</i> .....	50
3.2.1	Perancangan Program Arduino.....	53
3.2.2	Perancangan Program Aplikasi <i>Android</i> .....	61
<b>BAB IV PENGUJIAN DAN ANALISA DATA.....</b>		<b>69</b>
4.1	Pengujian Rangkaian <i>Power Supply</i> .....	69
4.2	Pengujian Sensor <i>Water Level</i> Funduino .....	70
4.3	Pengujian Sensor HSM 20-G.....	73
4.3.1	Pengujian Temperatur .....	73
4.3.2	Pengujian Kelembaban .....	76
4.4	Pengujian Sensor Cahaya .....	78
4.5	Pengujian Pompa Air dengan <i>Switching</i> TRIAC.....	80
4.6	Pengujian <i>Ultrasonic Humidifier</i> dengan <i>Switching Relay</i> .....	81
4.7	Pengujian LCD ( <i>Liquid Crystal Display</i> ) 20x4 .....	82
4.8	Pengujian RTC DS1307 .....	82



4.9 Pengujian Modul SD Card.....	83
4.10 Pengujian Modul <i>Bluetooth</i> HC-05.....	83
4.10.1 Pengujian <i>Bluetooth</i> Tanpa Penghalang.....	87
4.10.2 Pengujian <i>Bluetooth</i> Dengan Penghalang .....	92
4.11 Pengujian Aplikasi <i>Android</i> .....	98
4.12 Pengujian Keseluruhan Sistem Alat.....	101
<b>BAB V PENUTUP.....</b>	<b>105</b>
5.1 Kesimpulan .....	105
5.2 Saran .....	106
<b>DAFTAR PUSTAKA .....</b>	<b>107</b>
<b>LAMPIRAN A.....</b>	<b>109</b>
A.1. Program Arduino .....	109
A.2 Program MIT App Inventor.....	128
<b>LAMPIRAN B.....</b>	<b>131</b>
B.1 Pin Mapping Arduino Mega 2560 .....	131
B.2 HSM 20-G .....	134
B.3 Modul <i>Light Dependent Resistor</i> .....	136
B.2 Modul <i>Bluetooth</i> HC-05 .....	137
B.3 Konfigurasi Pin Modul <i>Bluetooth</i> HC-05 .....	138
B.4 Perintah AT <i>Command</i> .....	139
<b>LAMPIRAN C.....</b>	<b>145</b>
C.1 Dokumentasi .....	145
C.2 Penyemaian Benih Selada Hari ke-4 (Senin, 10 Juli 2017) .....	146
C.3. Penyemaian Benih Selada Hari ke-4 (Selasa, 11 Juli 2017) ...	147
C.4 Penyemaian Benih Selada Hari ke-5 (Rabu, 12 Juli 2017) .....	148
C.5 Penyemaian Benih Selada Hari ke-5 (Kamis, 13 Juli 2017) ....	149
C.6 Penyemaian Benih Selada Hari ke-6 (Sabtu, 15 Juli 2017) .....	151
<b>RIWAYAT HIDUP PENULIS.....</b>	<b>153</b>

-----Halaman ini sengaja dikosongkan-----

## DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Hidroponik Sistem <i>Wick</i> .....	8
Gambar 2.2 Penyemaian Benih Hidroponik menggunakan <i>Starter Kit</i> .....	9
Gambar 2.3 Diagram Blok Komunikasi Data .....	10
Gambar 2.4 Arduino Mega 2560 .....	12
Gambar 2.5 <i>Software</i> Arduino IDE .....	13
Gambar 2.6 Simbol dari TRIAC .....	14
Gambar 2.7 Rangkaian TRIAC Sebagai Saklar .....	14
Gambar 2.8 Diagram Blok <i>Power Supply</i> .....	15
Gambar 2.9 Trafo <i>Step Down</i> .....	15
Gambar 2.10 <i>Rectifier</i> pada <i>Power Supply</i> .....	16
Gambar 2.11 <i>Filter</i> atau Penyaring pada <i>Power Supply</i> .....	16
Gambar 2.12 Rangkaian Pembagi Tegangan pada <i>Power Supply</i> .....	17
Gambar 2.13 Rangkaian Sederhana <i>Power Supply</i> Keluaran 6 V DC ..	17
Gambar 2.14 Prinsip Kerja Sensor Kapasitif .....	19
Gambar 2.15 Bentuk Fisik LDR dan Simbolnya .....	20
Gambar 2.16 Rangkaian RTC DS1307 dengan Mikrokontroler .....	21
Gambar 2.17 Rangkaian <i>Driver Motor H-Bridge</i> .....	22
Gambar 2.18 Bagian dan Prinsip Kerja dari <i>Relay</i> .....	23
Gambar 2.19 Prinsip Kerja <i>Ultrasonic Humidifier</i> .....	24
Gambar 2.20 Prinsip Kerja Pompa Sentrifugal .....	27
Gambar 2.21 Tampilan Awal App Inventor .....	32
Gambar 3.1 Diagram Fungsional Sistem .....	33
Gambar 3.2 Rangkaian Modul <i>Shield</i> Arduino Mega 2560 .....	37
Gambar 3.3 Skematik Rangkaian SSR Menggunakan TRIAC .....	38
Gambar 3.4 Skematik Rangkaian <i>Power Supply</i> .....	39
Gambar 3.5 Skematik Rangkaian Pengondisi Sinyal Sensor HSM 20-G .....	40
Gambar 3.6 Konfigurasi Sensor Funduino .....	41
Gambar 3.7 Peletakan Sensor Funduino pada <i>Box Starter Kit</i> .....	41
Gambar 3.8 Rangkaian Modul RTC DS1307 .....	43
Gambar 3.9 Rangkaian <i>H-Bridge</i> pada <i>Driver Motor</i> L298N .....	44
Gambar 3.10 Rangkaian <i>Driver Motor</i> L298N .....	45

Gambar 3.11	Perancangan Desain <i>Hardware Plant</i> .....	46
Gambar 3.12	Perancangan <i>Driver Relay</i> .....	47
Gambar 3.13	Pengkabelan Modul <i>Bluetooth HC-05</i> dengan <i>Arduino</i> .....	49
Gambar 3.14	Pengaturan <i>AT Command</i> .....	50
Gambar 3.15	<i>Flowchart</i> Keseluruhan Sistem Alat.....	52
Gambar 3.16	<i>Flowchart</i> Komunikasi Data <i>Bluetooth</i> .....	53
Gambar 3.17	Deklarasi Variabel dan Tipe Data.....	54
Gambar 3.18	Inisialisasi pada <i>Void Setup</i> .....	55
Gambar 3.19	Fungsi Program HSM-20G.....	56
Gambar 3.20	Fungsi Program Modul RTC DS1307 .....	56
Gambar 3.21	Fungsi Program <i>SD Card</i> .....	57
Gambar 3.22	Fungsi Program Memasukkan Nilai <i>Set Point</i> untuk Temperatur Selada .....	58
Gambar 3.23	Program Menu Sesuai Baris dan Kolom.....	59
Gambar 3.24	Program Pengaktifan Aktuator .....	60
Gambar 3.25	Program Komunikasi Data <i>Bluetooth</i> dengan <i>Android</i> ..	60
Gambar 3.26	Program <i>Button</i> Aplikasi <i>Android</i> .....	61
Gambar 3.27	Tampilan <i>Designer View</i> .....	62
Gambar 3.28	Tampilan <i>Layout</i> Aplikasi <i>Android</i> .....	63
Gambar 3.29	Tampilan <i>Blocks View</i> .....	64
Gambar 3.30	Blok Program Inisialisasi Variabel Data .....	65
Gambar 3.31	Blok Program <i>Close Application</i> .....	65
Gambar 3.32	Blok Program <i>List Picker Bluetooth</i> .....	65
Gambar 3.33	Blok Program Konektivitas <i>Bluetooth</i> .....	65
Gambar 3.34	Blok Program Tampilan Waktu .....	66
Gambar 3.35	Blok Program Penerimaan Data .....	66
Gambar 3.36	Blok Program Menampilkan Nama Sayur .....	67
Gambar 3.37	Blok Program <i>Button</i> Pompa .....	68
Gambar 3.38	Blok Program <i>Button Humidifier</i> .....	68
Gambar 4.1	Metode Pengujian <i>Power Supply</i> .....	69
Gambar 4.2	Metode Pengujian Sensor Ketinggian Air .....	71
Gambar 4.3	Program Pengujian Sensor Ketinggian Air.....	72
Gambar 4.4	Metode Pengujian Sensor HSM 20-G .....	73
Gambar 4.5	Data Temperatur .....	74
Gambar 4.6	Data Kelembaban .....	76
Gambar 4.7	Metode Pengujian Sensor Cahaya .....	78
Gambar 4.8	Program Pengujian Sensor Cahaya.....	79
Gambar 4.9	Pengujian Pompa Air dengan <i>Switching SSR TRIAC</i> ....	81
Gambar 4.10	Pengujian <i>Ultrasonic Humidifier</i> dengan <i>Relay</i> .....	81

Gambar 4.11	Program Menampilkan Tulisan pada LCD 20x4 .....	82
Gambar 4.12	Pengujian LCD 20x4.....	82
Gambar 4.13	Pengujian RTC DS1307.....	83
Gambar 4.14	Pengujian Modul SD <i>Card</i> sebagai <i>Datalogger</i> .....	83
Gambar 4.15	Program Pengujian <i>Bluetooth</i> .....	84
Gambar 4.16	Ilustrasi Pengujian Tanpa Penghalang .....	84
Gambar 4.17	Ilustrasi Pengujian Dengan Penghalang .....	85
Gambar 4.18	Tampilan Aplikasi Ketika Sudah Terhubung.....	86
Gambar 4.19	Tampilan Aplikasi Ketika Menerima Data Berupa Angka.....	86
Gambar 4.20	Tampilan Aplikasi Ketika Tidak Terhubung .....	99
Gambar 4.21	Tampilan Aplikasi Ketika Menerima Data .....	99
Gambar 4.22	Tampilan Aplikasi Ketika Koneksi Terputus.....	100
Gambar 4.23	Tampilan Aplikasi Ketika <i>Humidifier</i> Aktif .....	100
Gambar 4.24	Tampilan Aplikasi Ketika Pompa Aktif.....	101
Gambar 4.25	Penyemaian Benih Selada Hari ke-1 (Sabtu, 8 Juli 2017).....	102

-----Halaman ini sengaja dikosongkan-----

## DAFTAR TABEL

	HALAMAN
Tabel 2.1	Data Parameter Sayuran.....8
Tabel 2.2	Spesifikasi Arduino Mega 2560..... 11
Tabel 2.3	Kelas <i>Bluetooth</i> .....28
Tabel 2.4	Perintah <i>AT Command</i> .....29
Tabel 2.5	Tingkatan Versi <i>Android</i> .....30
Tabel 3.1	<i>Mapping</i> Pin <i>Analog/Digital</i> yang digunakan.....35
Tabel 3.2	Konfigurasi Pin Modul Sensor LDR dengan Arduino .....42
Tabel 3.3	<i>Mapping</i> Pin <i>SD Card</i> dengan Modul Arduino .....47
Tabel 3.4	Spesifikasi Modul <i>Bluetooth</i> HC-05 .....48
Tabel 3.5	<i>Default Status</i> Modul <i>Bluetooth</i> HC-05 .....48
Tabel 3.6	Pin <i>Mapping</i> Modul <i>Bluetooth</i> HC-05 dengan Arduino saat Perintah <i>AT Command</i> .....49
Tabel 3.7	Pin <i>Mapping</i> Modul <i>Bluetooth</i> HC-05 dengan Arduino saat Melakukan Komunikasi Data .....50
Tabel 4.1	Data Pengujian <i>Power Supply</i> .....70
Tabel 4.2	Data Pengujian Sensor Ketinggian Air .....71
Tabel 4.3	Data Perbandingan Ketinggian Air .....72
Tabel 4.4	Data Pengujian Temperatur Sensor HSM 20-G .....73
Tabel 4.5	Data Perbandingan Temperatur .....75
Tabel 4.6	Data Pengujian Kelembaban Sensor HSM 20-G .....76
Tabel 4.7	Data Perbandingan Kelembaban .....77
Tabel 4.8	Pengukuran Sensor Cahaya.....78
Tabel 4.9	Data Perbandingan Cahaya .....80
Tabel 4.10	Pengujian <i>Bluetooth</i> Tanpa Penghalang ( <i>Data Bit 8, Stop Bit=1, Parity 0 (None)</i> ).....87
Tabel 4.11	Pengujian <i>Bluetooth</i> Tanpa Penghalang ( <i>Data Bit 8, Stop Bit=1, Parity 1 (Odd)</i> ) .....88
Tabel 4.12	Pengujian <i>Bluetooth</i> Tanpa Penghalang ( <i>Data Bit 8, Stop Bit=1, Parity 2 (Even)</i> ) .....89
Tabel 4.13	Pengujian <i>Bluetooth</i> Tanpa Penghalang ( <i>Data Bit 8, Stop Bit=2, Parity 0 (None)</i> ).....90
Tabel 4.14	Pengujian <i>Bluetooth</i> Tanpa Penghalang ( <i>Data Bit 8, Stop Bit=2, Parity 1 (Odd)</i> ) .....91

Tabel 4.15	Pengujian <i>Bluetooth</i> Tanpa Penghalang ( <i>Data Bit 8, Stop Bit=2, Parity 2 (Even)</i> ).....	91
Tabel 4.16	Pengujian <i>Bluetooth</i> Dengan Penghalang ( <i>Data Bit 8, Stop Bit=1, Parity 0 (None)</i> ) .....	93
Tabel 4.17	Pengujian <i>Bluetooth</i> Dengan Penghalang ( <i>Data Bit 8, Stop Bit=1, Parity 1 (Odd)</i> ).....	94
Tabel 4.18	Pengujian <i>Bluetooth</i> Dengan Penghalang ( <i>Data Bit 8, Stop Bit=1, Parity 2 (Even)</i> ).....	94
Tabel 4.19	Pengujian <i>Bluetooth</i> Dengan Penghalang ( <i>Data Bit 8, Stop Bit=2, Parity 0 (None)</i> ) .....	95
Tabel 4.20	Pengujian <i>Bluetooth</i> Dengan Penghalang ( <i>Data Bit 8, Stop Bit=2, Parity 1 (Odd)</i> ).....	96
Tabel 4.21	Pengujian <i>Bluetooth</i> Dengan Penghalang ( <i>Data Bit 8, Stop Bit=2, Parity 2 (Even)</i> ).....	97
Tabel 4.22	Pengujian Sistem pada Pukul 09.00-09.15 .....	102
Tabel 4.23	Pengujian Sistem pada Pukul 12.30-12.45 .....	102
Tabel 4.24	Pengujian Sistem pada Pukul 16.30-16.45 .....	103



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pada pertumbuhan sayuran hidroponik, salah satu tahap yang penting yaitu pada tahap penyemaian benih. Tahap penyemaian diawali dengan pecahnya benih menjadi tunas yang nantinya akan tumbuh menjadi bibit sayuran hidroponik. Disebut paling penting, karena jika saat penyemaian benih tidak dilakukan penanganan dengan benar maka akan berpengaruh pada pertumbuhan tanaman untuk tahap selanjutnya. Beberapa faktor yang dapat mempengaruhi tahap penyemaian benih diantaranya temperatur, kelembaban, kadar air dan intensitas cahaya. Keempat parameter tersebut perlu dikontrol sewaktu-waktu untuk menghasilkan bibit-bibit sayuran hidroponik yang siap tanam. Disamping itu, saat ini pemilik kebun sayur hidroponik masih melakukan pengontrolan keempat parameter tersebut secara manual. Hal ini dirasa kurang efektif dan efisien, karena pemilik kebun sayur harus melakukan pengontrolan secara langsung pada tahap penyemaian tersebut.

Berdasarkan kondisi tersebut, diperlukan suatu sistem yang dapat mengontrol temperatur, kelembaban, dan kadar air pada tanaman hidroponik, serta menjaga intensitas cahaya yang diterima tanaman. Selain itu, data-data dari sayuran ini dapat dilihat pada *Smartphone Android* dengan komunikasi data menggunakan modul *bluetooth* sehingga dapat memudahkan pemilik kebun sayur hidroponik untuk memonitoring dan mengetahui kondisi tanaman tanpa harus melakukan pengecekan secara langsung. Dengan adanya sistem ini, diharapkan dapat memudahkan pekerjaan pemilik kebun sayur untuk melakukan penanganan serta dapat meminimalisir banyaknya benih sayuran yang tidak siap tanam karena permasalahan yang ada pada kondisi lingkungan saat ini.

### **1.2 Permasalahan**

Proses penyemaian merupakan tahap yang penting, karena dalam tahap tersebut dipengaruhi oleh lingkungan yang berubah-ubah. Beberapa faktor lingkungan yang mempengaruhi sistem pertumbuhan sayuran hidroponik yaitu kadar air, temperatur, intensitas cahaya dan kelembaban. Parameter-parameter tersebut sangat rawan berubah karena perubahan kondisi lingkungan yang tidak menentu.

Untuk menggantikan pengontrolan secara manual, maka dibutuhkan suatu alat yang mampu mengontrol tanaman tersebut. Selain itu, pada umumnya, pemilik kebun sayur hidroponik masih melakukan pengontrolan secara manual pada tahap penyemaian, baik dalam hal pemberian air pada benih tanaman agar tetap tersuplai air maupun penanganan dalam faktor temperatur dan kelembaban tanaman mengingat temperatur udara di Surabaya berkisar hingga 33,8°C. Penanganan manual tersebut kurang efektif dan efisien. Untuk menggantikan pengontrolan secara manual, maka dibutuhkan mikrokontroler. Maka dari itu, pada Tugas Akhir ini, digunakan mikrokontroler Arduino Mega 2560 untuk mengontrol agar kadar air, temperatur, cahaya dan kelembaban tetap dalam kondisi sesuai dengan yang dibutuhkan oleh sayuran. Serta, dengan menggunakan aplikasi *Android*, pemilik kebun sayur dapat mengakses data-data parameter dan mengontrol benih sayur yang ditanam melalui komunikasi via *bluetooth*.

### **1.3 Batasan Masalah**

Pada tugas akhir ini, memiliki batasan-batasan masalah yang diambil, diantaranya :

1. Pengontrolan parameter yang digunakan yaitu temperatur, kelembaban, intensitas cahaya dan kadar air.
2. Sayuran yang dapat dijadikan sampel untuk ditanam pada alat ini yaitu pak coy, selada dan bayam.
3. Alat ini digunakan untuk tahap penyemaian saja dengan satu jenis sayuran untuk satu kali proses penyemaian.
4. *Smartphone* yang digunakan pada tahap pengujian yakni *smartphone android* versi *Lollipop*.

### **1.4 Tujuan**

Tujuan dari penelitian ini adalah :

1. Merancang sistem otomatis untuk menjaga temperatur, kelembaban, dan ketinggian air pada tahap penyemaian (Penanggung jawab Moh. Ilham Aziz W.)
2. Membuat aplikasi *android* untuk memonitoring dan mengontrol dari alat yang dibuat, serta komunikasi data via *bluetooth*. (Penanggung jawab Safitri Febrianti)

## 1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, tahap persiapan, tahap perencanaan dan pembuatan alat, tahap pengujian dan analisis, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur akan dipelajari mengenai identifikasi fisik, studi tentang Arduino dan karakteristik baik secara *hardware* dan *software*, mempelajari sensor HSM-20G, mempelajari sensor *Water Level*, mempelajari konsep TRIAC untuk mengaktifkan aktuator berupa *humidifier* dan pompa air, studi tentang pengkondisian sistem hidroponik, mempelajari *data logger* pada modul *SD Card*, mengintegrasikan data-data sensor dengan modul *SD Card* untuk mendapatkan *datalogger*, mempelajari konsep dasar komunikasi data, mempelajari komunikasi data menggunakan teknologi *bluetooth*, mempelajari modul *bluetooth* HC-05 beserta karakterisasi dan spesifikasinya, mengintegrasikan modul *bluetooth* dengan Arduino, konsep pemrograman pada sistem operasi *Android*, mempelajari pemrograman App Inventor guna pembuatan aplikasi *Android*. Pada tahap perancangan sistem dan perancangan alat, sensor HSM 20G, sensor cahaya LDR, dan sensor ketinggian air yang telah dibuat dikalibrasi agar dapat dibaca oleh Arduino, pengaturan pada modul *bluetooth* HC-05, membuat aplikasi untuk *smartphone Android* guna menampilkan parameter sayuran serta guna monitoring alat. Pada tahap pengujian dan analisis, dilakukan pengujian pada sensor-sensor yang digunakan, pengujian aktuator, pengujian modul *bluetooth* dan aplikasi yang telah dibuat, serta pengujian pada keseluruhan sistem. Data percobaan yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

## 1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

### Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan dan relevansi.

## **Bab II Teori Dasar**

Bab ini menjelaskan tentang tinjauan pustaka dari hidroponik, Arduino Mega 2560, TRIAC *Switching*, *Power Supply*, Sensor HSM-20G, Sensor Ketinggian Air, Sensor Cahaya LDR, RTC (*Real Time Clock*), *Driver Motor*, *Relay*, *Ultrasonic Humidifier*, Modul SD Card, LCD 20x4 dan Pompa Air, teori komunikasi data, teori *bluetooth*, sistem operasi *Android*, *software* Arduino IDE, *software* App Inventor.

## **Bab III Perancangan Sistem**

Bab ini membahas perencanaan dan pembuatan perangkat keras (*Hardware*) yang meliputi desain alat serta pengimplementasian sensor dan aktuator yang digunakan, pengaturan modul *bluetooth*, dan pembuatan perangkat lunak (*Software*) yang meliputi program pada Arduino IDE untuk menjalankan alat tersebut, serta pembuatan program untuk aplikasi Android pada MIT App Inventor.

## **Bab IV Pengujian dan Analisis**

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian sensor temperatur, kelembaban dan ketinggian air, pengujian rangkaian TRIAC untuk mengaktifkan pompa air, pengujian rangkaian *relay* untuk mengaktifkan *humidifier*, serta pengujian keseluruhan alat terhadap tanaman hidroponik. Selain itu, dilakukan pengujian *bluetooth*, dan pengujian aplikasi *Android*.

## **Bab V Penutup**

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

### **1.7 Relevansi**

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dapat mempermudah pekerjaan petani hidroponik dalam proses penyema-

ian guna mengetahui kondisi benih sayuran serta dapat meminimalisir kerusakan benih sayuran karena masalah cuaca dan perubahan kondisi lingkungan.

-----Halaman ini sengaja dikosongkan-----

## **BAB II**

### **TEORI DASAR**

Bab ini membahas mengenai teori dasar dari peralatan yang digunakan dalam Tugas Akhir yang berjudul Sistem Otomasi Penyemaian Benih Sayuran Hidroponik pada Kebun Sayur Surabaya yang meliputi 19 poin. Penjelasan dari teori dasar ini meliputi terbagi menjadi 2 bagian pada masing-masing mahasiswa, Moh. Ilham Aziz mencari bahan materi pada poin 2.1 sampai dengan poin 2.9, sedangkan Safitri Febrianti mencari bahan materi pada poin 2.10 sampai dengan poin 2.19. Materi ini digunakan sebagai dasar materi untuk pembuatan alat yang dibuat masing-masing mahasiswa untuk pembuatan keseluruhan alat ini.

#### **2.1 Hidroponik**

Pada sub bab ini akan dijelaskan tentang teori dasar mengenai hidroponik, teknik atau sistem yang terdapat pada pola cocok tanam hidroponik, sayuran yang dapat ditanam dan dijadikan sampel pada pengujian alat, serta penjelasan mengenai sayuran selada.

##### **2.1.1 Teori Dasar Hidroponik [1]**

Hidroponik adalah budidaya menanam dengan memanfaatkan air tanpa menggunakan tanah dengan menekankan pada pemenuhan kebutuhan nutrisi bagi tanaman. Kebutuhan air pada hidroponik lebih sedikit daripada kebutuhan air pada budidaya dengan tanah. Budidaya tanaman secara hidroponik memiliki beberapa keuntungan dibandingkan dengan budidaya secara konvensional, yaitu pertumbuhan tanaman dapat dikontrol, tanaman dapat berproduksi dengan kualitas dan kuantitas yang tinggi, tanaman jarang terserang hama penyakit karena terlindungi, pemberian air irigasi dan larutan hara lebih efisien dan efektif, dapat diusahakan terus menerus tanpa tergantung oleh musim, dan dapat diterapkan pada lahan yang sempit.

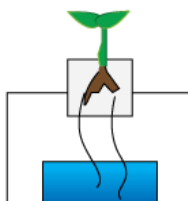
Pada proses pertumbuhan sayuran hidroponik terdapat tahap penyemaian benih. Tahap ini merupakan tahap awal dan menjadi tahap yang paling rawan karena sangat menentukan kualitas sayuran.

Dalam penyemaian ini, juga diperhatikan dalam hal temperatur dan kelembaban dari media tanam. Temperatur udara mempengaruhi kecepatan pertumbuhan maupun sifat dan struktur tanaman. Tumbuhan dapat tumbuh dengan baik pada temperatur

optimum. Untuk tumbuhan daerah tropis temperatur optimumnya berkisar 22°C-37°C. Tetapi temperatur ini sangat dipengaruhi oleh jenis dan fase pertumbuhan tanaman.

### 2.1.2 Hidroponik Sistem Wick [1]

Terdapat beberapa macam teknik yang dapat diterapkan pada sistem hidroponik. Salah satunya yaitu dengan sistem *Wick*. Sistem *Wick* paling banyak digunakan oleh pemula karena metode nya yang sederhana. Sistem ini termasuk pasif dan air dari dalam wadah / bak mengalir ke dalam media pertumbuhan melalui sejenis sumbu. Sistem *Wick* bekerja dengan baik untuk tanaman dengan ukuran kecil dan dapat diterapkan pada tahap penyemaian. Pada sistem ini, tiap net pot diisi media tanam seperti *rockwool* dan sumbu yang menjulur ke bawah yang berfungsi untuk menyerap larutan ke akar tanaman. Gambar 2.1 merupakan ilustrasi dari penanaman tanaman hidroponik menggunakan sistem *Wick*.



**Gambar 2.1** Hidroponik Sistem *Wick*

### 2.1.3 Sayuran Hidroponik [2]

Tidak semua jenis sayuran bisa ditanam menggunakan sistem hidroponik. Sayuran yang bisa ditanam biasanya merupakan sayuran hijau. Beberapa jenis sayuran tersebut antara lain selada, pakcoy dan bayam. Ketiga sayuran ini memiliki kadar temperatur dan kelembaban masing-masing.

Berikut ini merupakan data temperatur dan kelembaban sayuran :

**Tabel 2.1** Data Parameter Sayuran

Nama Sayuran	Temperatur	Kelembaban
Selada	27°C	±80%
Pakcoy	29°C	±75%
Bayam	29°C	±75%





**Gambar 2.2** Penyemaian Benih Hidroponik menggunakan *Starter Kit*

#### **2.1.4 Selada (*Lactuca sativa* L.) [3]**

Selada (*Lactuca sativa* L.) merupakan sayuran daun yang berumur semusim dan termasuk dalam famili *Compositae*. Menurut jenisnya ada yang dapat membentuk krop dan ada pula yang tidak. Jenis yang tidak membentuk krop daun-daunnya berbentuk “*rosette*”. Warna daun selada hijau terang sampai putih kekuningan.

Selada dapat tumbuh pada dataran rendah maupun dataran tinggi (pegunungan). Namun, akan tumbuh baik di dataran tinggi (pegunungan). Di dataran rendah kropnya kecil-kecil dan cepat berbunga. Temperatur ideal untuk pertumbuhan selada adalah 15-27°C. Temperatur yang lebih tinggi dari 30°C dapat menghambat pertumbuhan, merangsang tumbuhnya tangkai bunga, dan dapat menyebabkan rasa pahit.

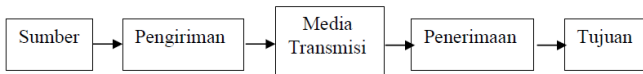
### **2.2 Teori Komunikasi Data [4]**

Komunikasi data merupakan proses pertukaran data atau pengiriman data dari sumber ke tujuan. Beberapa hal yang penting dalam melakukan komunikasi data adalah jenis komunikasi yang digunakan, misalnya menggunakan kabel, nirkabel maupun frekuensi radio. Adapun tujuan dari komunikasi data ini adalah :

- a. Memungkinkan pengiriman data dalam jumlah yang besar, efisien, tanpa kesalahan dan ekonomis dari satu tempat ke tempat yang lain.
- b. Memungkinkan penggunaan sistem komputer dan peralatannya yang jauh.
- c. Memungkinkan penggunaan sistem komputer secara terpusat maupun secara tersebar.

- d. Mempermudah kemungkinan pengolahan dan pengaturan data yang ada dalam berbagai macam sistem komputer.
- e. Mengurangi waktu untuk pengolahan data.
- f. Mendapatkan data langsung dari sumbernya.
- g. Mempercepat penyebaran informasi

Secara umum, diagram blok komunikasi data terdapat pada Gambar 2.3 :



**Gambar 2.3** Diagram Blok Komunikasi Data

Penjelasan dari diagram blok pada Gambar 2.3 yaitu terdapat sumber yang merupakan masukan data atau informasi yang akan dikirimkan ke tujuan, yang kemudian melalui proses pengiriman data dari sumber ke media transmisi. Media transmisi yang dimaksud yaitu jalur transmisi yang menghubungkan antara sumber dan tujuan. Selanjutnya melalui proses penerimaan data yang kemudian dikirimkan ke tujuan. Pada tujuan, akan ditampilkan hasil data yang dikirim oleh sumber.

Dalam komunikasi data jenis komunikasi ada tiga macam yaitu :

- a. *Simplex*, yaitu komunikasi data hanya dengan satu arah dari sumber ke tujuan.
- b. *Half duplex*, yaitu komunikasi data dengan dua arah tetapi tidak bisa melakukan komunikasi secara bersamaan.
- c. *Full duplex*, yaitu komunikasi data dengan dua arah dan bisa melakukan komunikasi secara bersamaan.

## 2.3 Arduino Mega 2560 [5]

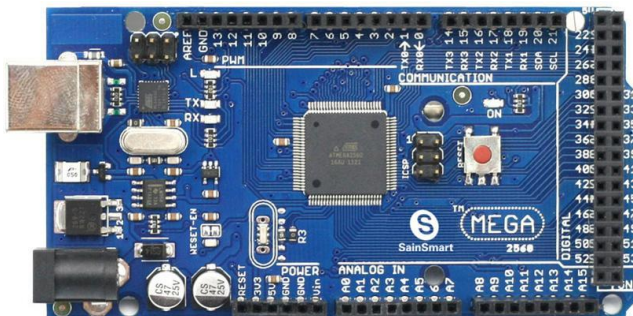
Arduino merupakan perangkat elektronik atau papan rangkaian elektronik *open – source* yang di dalamnya terdapat komponen utama, yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah *chip* atau IC (*Integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Jadi

mikrokontroler bertugas sebagai otak yang mengendalikan *input*, proses, dan *output* sebuah rangkaian elektronik.

Tipe Arduino yang akan digunakan pada penelitian kali ini yaitu Arduino Mega 2560. Arduino Mega 2560 adalah mikrokontroler berbasis ATmega2560 dengan *Clock Speed* 16Mhz dan *Flash Memory* 256KB. Tegangan operasi untuk arduino jenis ini yaitu 5 V. Sedangkan tegangan *input* yang direkomendasikan yakni 7 – 12 V. Arduino ini memiliki 54 pin digital *input/output* pada pin 22-53 dengan 15 pin diantaranya merupakan pin PWM pada pin 0-13, 16 pin analog *input* pada pin A0 – A15, sambungan USB, sambungan catu daya tambahan dan tombol pengaturan ulang. Pada Tabel 2.2 merupakan spesifikasi dari Arduino Mega 2560.

**Tabel 2.2** Spesifikasi Arduino Mega 2560

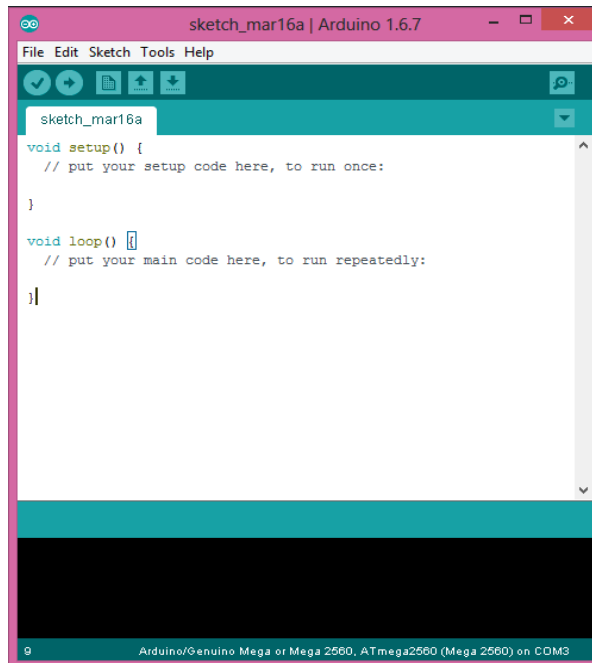
<b>Spesifikasi</b>	<b>Keterangan</b>
<i>Chip</i> Mikrokontroler	ATmega2560
Tegangan Operasi	5 V
Tegangan <i>Input</i> (Rekomendasi)	7 V-12 V
Tegangan <i>Input</i> (Limit)	6 V- 20 V
<i>Pin</i> Digital I/O	54, (15 buah diantaranya dapat digunakan sebagai <i>Output</i> PWM)
<i>Pin</i> Analog <i>Input</i>	16 (A0 – A.15)
Arus DC per <i>Pin</i> I/O	40 mA
Arus DC <i>Pin</i> 3,3V	50 mA
Memori <i>Flash</i>	256 KB, 8 KB telah digunakan untuk <i>Bootloader</i>
SRAM	8 Kb
EEPROM	4 Kb
<i>Clock Speed</i>	16 MHz



**Gambar 2.4** Arduino Mega 2560

## 2.4 Arduino IDE[5]

Arduino IDE adalah perangkat lunak yang digunakan untuk memprogram, monitoring dan *debugging* mikrokontroler Arduino. IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler. Pada Gambar 2.5 merupakan tampilan awal untuk membuat program pada *software* Arduino IDE.



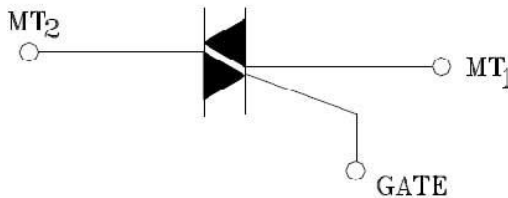
**Gambar 2.5** Software Arduino IDE

## 2.5 TRIAC Switching[6]

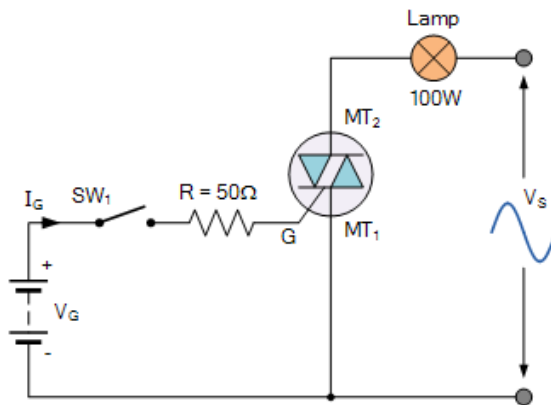
TRIAC adalah perangkat semikonduktor berterminal tiga yang berfungsi sebagai pengendali arus listrik. Nama TRIAC ini merupakan singkatan dari *Triode For Alternating Current* (Trioda untuk arus bolak balik). Sama seperti SCR, TRIAC juga tergolong sebagai *Thyristor* yang berfungsi sebagai pengendali atau *Switching*. Namun, berbeda dengan SCR yang hanya dapat dilewati arus listrik dari satu arah (*unidirectional*), TRIAC memiliki kemampuan yang dapat mengalirkan arus listrik ke kedua arah (*bidirectional*) ketika dipicu. Terminal *Gate* TRIAC hanya memerlukan arus yang relatif rendah untuk dapat mengendalikan aliran arus listrik AC yang tinggi dari dua arah terminalnya. TRIAC sering juga disebut dengan *Bidirectional Triode Thyristor*.

Pada dasarnya, sebuah TRIAC sama dengan dua buah SCR yang disusun dan disambungkan secara antiparalel (paralel yang berlawanan arah) dengan terminal gerbang nya dihubungkan bersa-

ma menjadi satu. Jika dilihat dari strukturnya, TRIAC merupakan komponen elektronika yang terdiri dari 4 lapis semikonduktor dan 3 Terminal, ketiga Terminal tersebut diantaranya adalah MT1, MT2 dan *Gate*. MT adalah singkatan dari *Main Terminal*. Pada Gambar 2.6 merupakan symbol dari TRIAC, dan pada Gambar 2.7 merupakan aplikasi TRIAC sebagai saklar.



**Gambar 2.6** Simbol dari TRIAC

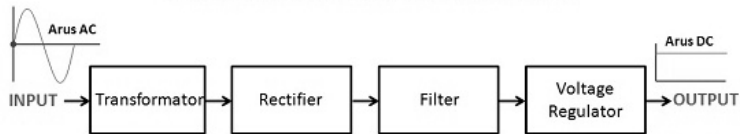


**Gambar 2.7** Rangkaian TRIAC Sebagai Saklar

## 2.6 Power Supply [6]

*Power Supply* atau pencatu daya merupakan rangkaian elektronika yang dapat menghasilkan energi listrik atau sebagai sumber energi untuk rangkaian elektronika lainnya. Sumber arus dari *power supply* adalah arus bolak – balik (AC) dari pembangkit listrik yang kemudian diubah menjadi arus searah (DC). Untuk dapat melakukan hal tersebut *power supply* memerlukan perangkat yang bisa mengubah arus AC menjadi DC. Sebuah DC *power supply* pada dasarnya memiliki 4 bagian utama agar dapat menghasilkan arus DC

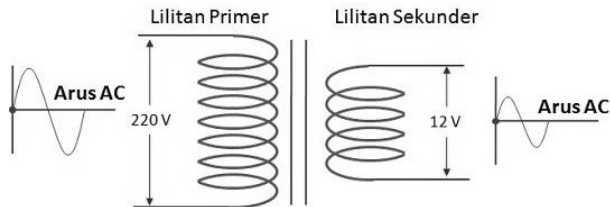
yang stabil. Keempat bagian utama tersebut diantaranya adalah Transformator, *rectifier*, *filter* dan *voltage regulator*. Berikut adalah diagram blok *power supply*.



**Gambar 2.8** Diagram Blok *Power Supply*

Berikut ini adalah penjelasan singkat tentang prinsip kerja DC *power supply* pada masing-masing blok berdasarkan diagram blok diatas.

a. Transformator (Trafo)



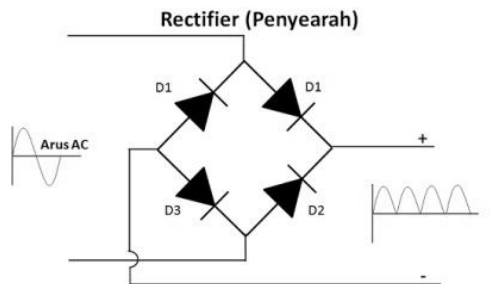
**Gambar 2.9** Trafo *Step Down*

Transformator yang digunakan untuk DC *power supply* adalah Transformer jenis *Step-down* yang berfungsi untuk menurunkan tegangan listrik sesuai dengan kebutuhan komponen Elektronika yang terdapat pada rangkaian adaptor (DC *power supply*). Transformator bekerja berdasarkan prinsip Induksi elektromagnetik yang terdiri dari 2 bagian utama yang berbentuk lilitan yaitu lilitan primer dan lilitan sekunder. Lilitan primer merupakan *input* dari transformator sedangkan *output*-nya pada lilitan sekunder. Meskipun tegangan telah diturunkan, *output* dari transformator masih berbentuk arus bolak-balik (arus AC) yang harus diproses selanjutnya.

b. *Rectifier* (Penyearah Gelombang)

*Rectifier* atau penyearah gelombang adalah rangkaian elektronika dalam *power supply* yang berfungsi untuk

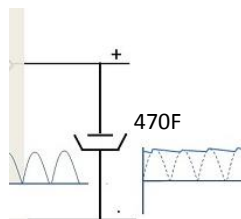
mengubah gelombang AC menjadi gelombang DC setelah tegangannya diturunkan oleh transformator *step down*. Rangkaian *Rectifier* biasanya terdiri dari komponen dioda. Terdapat 2 jenis rangkaian *Rectifier* dalam *power supply* yaitu “*Half Wave Rectifier*” yang hanya terdiri dari 1 komponen dioda dan “*Full Wave Rectifier*” yang terdiri dari 2 atau 4 komponen dioda.



**Gambar 2.10** *Rectifier* pada *Power Supply*

c. *Filter* (Penyaring)

Dalam rangkaian *power supply* (Adaptor), *filter* digunakan untuk meratakan sinyal arus yang keluar dari *Rectifier*. *Filter* ini biasanya terdiri dari komponen kapasitor (kondensator) yang berjenis elektrolit atau ELCO (*Electrolyte Capacitor*).



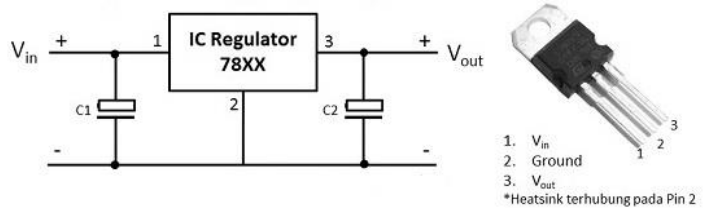
**Gambar 2.11** *Filter* atau Penyaring pada *Power Supply*

d. Pengatur Tegangan (*Voltage Regulator*)

Untuk menghasilkan tegangan dan arus DC (arus searah) yang tetap dan stabil, diperlukan *voltage regulator* yang berfungsi untuk mengatur tegangan

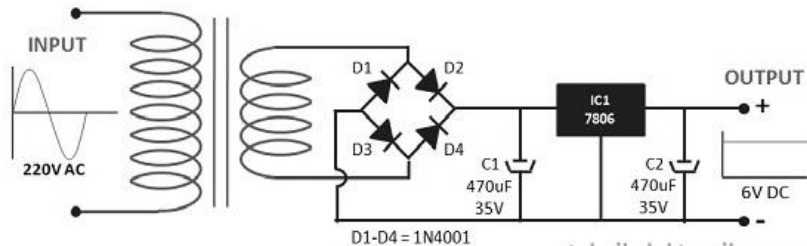


sehingga tegangan *output* tidak dipengaruhi oleh suhu, arus beban dan juga tegangan *input* yang berasal *output filter*. *Voltage regulator* pada umumnya terdiri dari Dioda Zener, Transistor atau IC (*Integrated Circuit*). Pada DC *power supply* yang canggih, biasanya *voltage regulator* juga dilengkapi dengan *short circuit protection* (perlindungan atas hubung singkat), *current limiting* (pembatas Arus) ataupun *over voltage protection* (perlindungan atas kelebihan tegangan).



**Gambar 2.12** Rangkaian Pembagi Tegangan pada *Power Supply*

Berikut ini adalah contoh rangkaian sederhana *Power Supply* dengan keluaran 6V:



**Gambar 2.13** Rangkaian Sederhana *Power Supply* Keluaran 6 V DC

## 2.7 Sensor HSM 20-G[7]

Sensor HSM-20G merupakan sensor pengukur temperatur dan kelembaban. Dalam penelitian ini, kemampuan sensor HSM 20-G yang dibutuhkan adalah respon sensor terhadap kehadiran objek pada temperatur dan kelembaban tertentu. Respon tersebut berupa perubahan resistansi sensor yang menghasilkan tegangan keluaran. Untuk mengetahui tegangan keluaran sensor, dibutuhkan tegangan DC sebesar 5 V. Batas tegangan keluaran dari sensor ini yaitu sekitar 1-3 V. Terdapat 4 pin pada sensor HSM-20G ini diantaranya pin untuk

VCC, pin GND, pin untuk keluaran temperatur, dan pin untuk keluaran kelembaban. Sensor ini dapat beroperasi untuk mendeteksi adanya temperatur dengan rentang antara 0-50°C, dan kelembaban dengan rentang antara 20-90% RH.

Untuk kelembaban, keluaran sensor ini berupa tegangan, sedangkan untuk temperatur, keluaran sensor berupa resistansi sehingga dibutuhkan rangkaian tambahan berupa pembagi tegangan guna mengetahui tegangan keluaran nya. Rangkaian tambahan untuk sensor ini akan dijelaskan pada bab III.

Prinsip kerja dari keluaran temperatur dari sensor ini menggunakan prinsip resistif yaitu perubahan besaran yang diindera. Contoh sensor yang bersifat resistif adalah RTD (*Resistance Thermal Detector*). RTD dibuat dari bahan kawat tahan korosi, kawat tersebut dililitkan pada bahan keramik isolator. Bahan kawat untuk RTD tersebut antara lain; platina, emas, perak, nikel dan tembaga, dan yang terbaik adalah bahan platina karena dapat digunakan menyensor temperatur sampai 1500°C. Keluaran temperatur HSM-20G berupa resistif dan sekelas jenisnya seperti RTD. Prinsip kerja RTD adalah ketika temperatur elemen RTD meningkat, maka resistansi elemen tersebut juga akan meningkat. Dengan kata lain, kenaikan temperatur logam yang menjadi elemen resistor RTD berbanding lurus dengan resistansinya. Spesifikasi RTD yang paling umum adalah 100  $\Omega$  (RTD PT100), yang berarti bahwa pada temperatur 0<sup>0</sup> C, elemen RTD harus menunjukkan nilai resistansi 100  $\Omega$ .

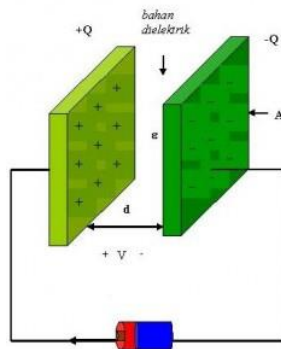
Dalam prakteknya, arus listrik akan mengalir melalui elemen RTD (elemen resistor) yang terletak pada tempat atau daerah yang mana temperaturnya akan diukur. Nilai resistansi dari RTD kemudian akan diukur oleh instrumen alat ukur, yang kemudian memberikan hasil bacaan dalam temperatur yang tepat, pembacaan temperatur ini didasarkan pada karakteristik resistansi yang diketahui dari RTD.

## **2.8 Sensor Ketinggian Air[7]**

Sensor ketinggian air berfungsi untuk mengukur ketinggian air. Pada umumnya sensor ketinggian air bekerja pada prinsip kapasitif. Sensor yang bekerja secara kapasitif yaitu berdasarkan perubahan muatan energi listrik yang dapat disimpan oleh sensor akibat perubahan jarak lempeng, perubahan luas penampang dan perubahan volume dielektrikum sensor kapasitif tersebut. Terdapat

beberapa sifat sensor kapasitif yang dimanfaatkan dalam pengukuran yaitu:.

- Jika luas permukaan dan dielektrika (udara) dijaga dalam kondisi konstan, maka perubahan nilai kapasitansi ditentukan oleh jarak antara kedua lempeng logam.
- Jika luas permukaan dan jarak kedua lempeng logam dijaga konstan dan volume dielektrum dapat dipengaruhi maka perubahan kapasitansi ditentukan oleh volume atau ketinggian cairan elektrolit yang diberikan.
- Jika jarak dan dielektrikum (udara) dijaga konstan, maka perubahan kapasitansi ditentukan oleh permukaan kedua lempeng logam yang saling berdekatan.



**Gambar 2.14** Prinsip Kerja Sensor Kapasitif

Konstruksi sensor kapasitif yang digunakan berupa dua buah lempeng logam yang diletakkan sejajar dan saling berhadapan. Jika diberi beda tegangan antara kedua lempeng logam tersebut, maka akan timbul kapasitansi antara kedua logam tersebut. Nilai kapasitansi yang ditimbulkan berbanding lurus dengan luas permukaan lempeng logam, berbanding terbalik dengan jarak antara kedua lempeng dan berbanding lurus dengan zat antara kedua lempeng tersebut (dielektrika), seperti ditunjukkan oleh persamaan berikut :

$$C = \epsilon_0 \epsilon_r \frac{A}{d} \dots\dots\dots(2.1)$$

Dimana:

$\epsilon_0$  : permitivitas ruang hampa

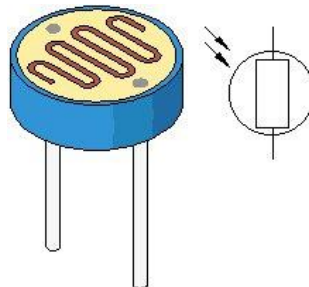
$A$  : luas plat atau lempeng dalam  $m^2$

d : jarak antara plat atau lempeng dalam  $m$

## 2.9 Sensor Cahaya LDR[8]

Sensor LDR berupa resistor yang dapat mengalami perubahan resistansi apabila mengalami perubahan penerimaan cahaya. Sensor ini terbuat dari *cadmium sulfide* yaitu merupakan bahan semikonduktor yang resistansinya berubah – ubah menurut banyaknya cahaya sinar yang mengenainya. Perlu diketahui bahwa nilai resistansi LDR sangat dipengaruhi oleh intensitas cahaya. Semakin banyak cahaya yang mengenainya, maka semakin menurun nilai resistansinya. Sebaliknya, jika cahaya yang mengenainya sedikit (gelap), maka nilai hambatannya menjadi semakin besar, sehingga arus listrik yang mengalir akan terhambat.

Fungsi LDR adalah sebagai saklar otomatis berdasarkan cahaya. Jika cahaya yang diterima oleh LDR banyak, maka nilai resistansi LDR akan menurun, dan listrik dapat mengalir (*ON*). Sebaliknya, jika cahaya yang diterima LDR sedikit, maka nilai resistansi LDR akan menguat, dan aliran listrik terhambat (*OFF*). LDR kerap difungsikan sebagai sebuah sensor cahaya dalam berbagai macam rangkaian elektronika seperti lampu penerangan jalan otomatis, lampu kamar tidur otomatis, dan sebagainya. Prinsip kerja LDR seperti variabel resistor pada umumnya. LDR dipasang pada sebuah rangkaian elektronika dan dapat memutus dan menyambung aliran listrik berdasarkan cahaya. Semakin banyak cahaya yang mengenainya, maka semakin menurun nilai resistansinya. Sebaliknya, jika cahaya yang mengenainya sedikit (gelap), maka nilai hambatannya menjadi semakin besar.

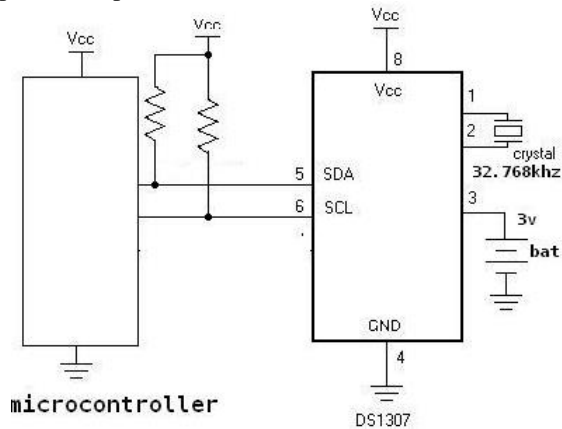


**Gambar 2.15** Bentuk Fisik LDR dan Simbolnya

### 2.10 RTC (*Real Time Clock*)[8]

RTC adalah sebuah rangkaian elektronika yang berfungsi sebagai acuan waktu. Pada umumnya RTC digunakan pada alat elektronika yang membutuhkan akurasi waktu yang sesuai dengan waktu dunia. RTC berbeda dengan jam biasa karena RTC umumnya hanya dalam bentuk IC. Dalam penggunaannya, dengan adanya RTC sebuah sistem dapat fokus dengan tugas utamanya. Selain itu, RTC mempunyai sumber daya yang berbeda dari sistem. Sehingga ketika sistem dimatikan RTC masih berfungsi dan waktunya tidak akan berhenti atau *ter – reset* saat *restart*. Data – data yang tersimpan pada IC DS3207 disimpan pada *register* 00H untuk detik, 01H untuk menit, 02H untuk jam, 03H untuk hari, 04H untuk tanggal, 05H untuk bulan 06H untuk tahun, 07H untuk kontrol dan RAM 56x8 pada *register* 08H – 3FH. *Register* tersebut bisa diakses oleh mikrokontroler melalui bus I2C.

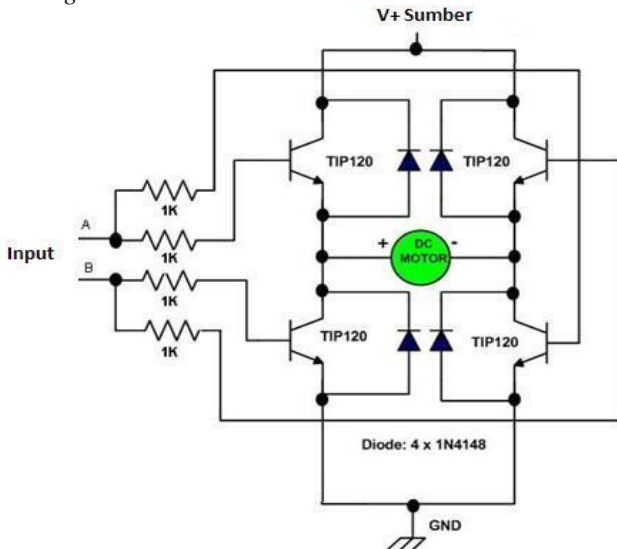
Pada pin 1 dan pin 2 dihubungkan dengan kristal eksternal 32,768 KHz. Pin 3 Vbat di hubungkan dengan baterai eksternal 3V. Pin 4 GND dapat kita hubungkan dengan *ground* pada baterai eksternal. Pin 8 Vcc di hubungkan ke Vcc minimum sistem. Pada pin 5, 6, dan 7 adalah sinyal data dari RTC ke mikrokontroler, di antaranya pin 7 adalah SQM, pin 6 adalah SCL dan pin 5 adalah SDA dan di masing-masing sinyal data tersebut kita berikan resistor 1k $\Omega$  hingga 10k $\Omega$  sebagai *pull up*, kecuali pin 7 yaitu SQM. Untuk lebih jelasnya dapat dilihat pada Gambar 2.16 berikut:



**Gambar 2.16** Rangkaian RTC DS1307 dengan Mikrokontroler

### 2.11 Driver Motor[9]

Rangkaian *driver* motor DC *H-Bridge* transistor ini dapat mengendalikan arah putaran motor DC dalam 2 arah dan dapat dikontrol dengan metode PWM (*Pulse Width Modulation*) maupun metode sinyal logika dasar TTL (*High*) dan (*Low*). Untuk pengendalian motor DC dengan metode PWM maka dengan rangkaian *driver* motor DC ini kecepatan putaran motor DC dapat dikendalikan dengan baik. *Driver* motor DC dengan metode logika TTL (0 dan 1) atau *High* dan *Low* hanya dapat mengendalikan arah putar motor DC dalam 2 arah tanpa pengendalian kecepatan putaran (kecepatan maksimum). Untuk mengendalikan motor DC dalam 2 arah dengan rangkaian *driver* motor DC *H-Bridge* diatas konfigurasi kontrol pada jalur *input* adalah dengan memberikan *input* berupa logika TTL ke jalur *input* A dan B. Untuk mengendalikan arah putar searah jarum jam adalah dengan memberikan logika TTL 1 (*high*) pada jalur *input* A dan logika TTL 0 (*low*) pada jalur *input* B. Untuk mengendalikan arah putar berlawanan arah jarum jam adalah dengan memberikan logika TTL 1 (*high*) pada jalur *input* B dan logika TTL 0 (*low*) pada jalur *input* A. Gambar 2.17 merupakan rangkaian *driver* motor *H-Bridge*.

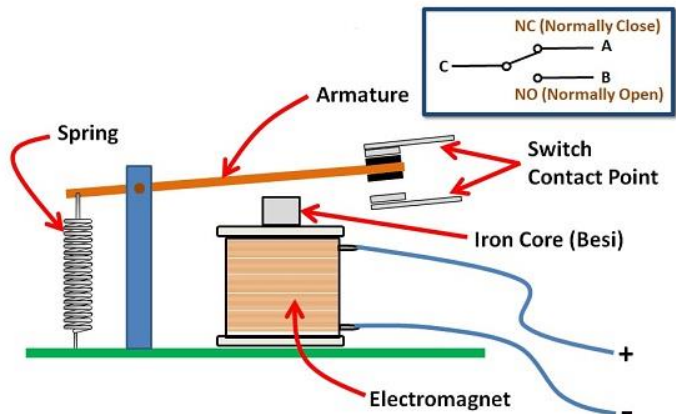


**Gambar 2.17** Rangkaian *Driver* Motor *H-Bridge*

## 2.12 Relay [9]

Rangkaian *relay* adalah rangkaian elektronika yang digunakan untuk mengendalikan sesuatu dari jarak jauh. *Relay* sendiri merupakan saklar magnetis paling sering digunakan pada setiap rangkaian elektronika. Dalam dunia elektronika, *relay* sangat berperan penting dalam suatu rangkaian beban arus tinggi dengan arus rendah. *Relay* menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan *relay* yang menggunakan elektromagnet 5V dan 50 mA mampu menggerakkan *armature relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A.

Pada dasarnya, *relay* terdiri dari 4 komponen dasar yaitu : *Electromagnet (Coil)*, *Armature*, *Switch Contact Point* (Saklar) dan *Spring*. Gambar 2.18 merupakan gambar bagian-bagian dari *relay*:



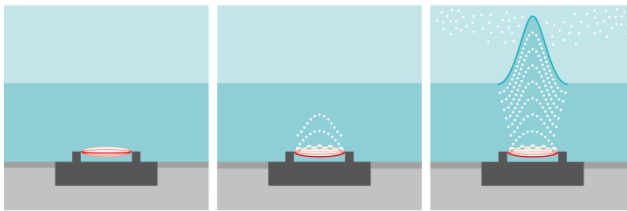
**Gambar 2.18** Bagian dan Prinsip Kerja dari *Relay*

## 2.13 Ultrasonic Humidifier [10]

*Ultrasonic humidifier* merupakan sebuah alat untuk menambah jumlah air di udara dalam suatu ruangan atau aliran udara. Fungsi utama dari *ultrasonic humidifier* yaitu untuk mempertahankan suatu ruangan agar tetap dalam keadaan lembab. Cara kerja dari alat ini yaitu ketika diafragma ultrasonik bergetar pada frekuensi

yang sangat tinggi akan memecah air menjadi tetesan halus yang keluar sebagai kabut dingin dan dapat mengubahnya kembali menjadi uap air. *Humidifier* terdiri dari dua lingkaran cakram keramik *piezoelektrik* yang dilengkapi arus melalui dua elektroda nikel. Karena keramik adalah *piezoelektrik*, *humidifier* beresilasi dengan adanya arus listrik pada frekuensi ultrasonik. Perangkat kemudian memancarkan kabut halus ini ke udara, secara instan mendistribusikan kelembaban ke seluruh ruangan.

Dari Gambar 2.19 dapat dijelaskan gambar paling kiri adalah ketika *humidifier* dalam keadaan nyala dan amplitudo dari osilator positif. Pada gambar bagian tengah adalah dimana osilator dalam keadaan negatif dan inersia pada air menciptakan ruang yang kosong. Pada gambar paling kanan adalah keadaan dimana nilai amplitudo sebesar kira-kira sebesar 10 *ultrasonic humidifier* akan mencapai 100% dari keluarannya.



**Gambar 2.19** Prinsip Kerja *Ultrasonic Humidifier*

## 2.14 Modul SD Card[11]

*MicroSD/SDCard* adalah kartu memori *non-volatile* yang dikembangkan oleh *SD Card Association* yang digunakan dalam perangkat portabel. Saat ini, teknologi *microSD* sudah digunakan oleh lebih dari 400 merek produk serta dianggap sebagai *standar industryde-facto*.

Keluarga *microSD* yang lain terbagi menjadi SDSC yang kapasitas maksimum resminya sekitar 2GB, meskipun beberapa ada yang sampai 4GB. SDHC (*High Capacity*) memiliki kapasitas dari 4GB sampai 32GB. Dan SDXC (*Extended Capacity*) kapasitasnya di atas 32GB hingga maksimum 2TB. Keberagaman kapasitas seringkali membuat kebingungan karena masing-masing protokol komunikasi sedikit berbeda.

Dari sudut pandang perangkat, semua kartu ini termasuk kedalam keluarga SD. *SD adapter* memungkinkan konversi fisik



kartu SD yang lebih kecil untuk bekerja di slot fisik yang lebih besar dan pada dasarnya ini adalah alat pasif yang menghubungkan pin dari *microSD* yang kecil ke pin adaptor *microSD* yang lebih besar.

### **2.15 *Liquid Cristal Display (LCD) 20x4* [11]**

LCD (*Liquid Cristal Display*) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD yang dipakai merupakan tipe berkarakter 20x4 baris, yang dapat menampilkan 20 karakter dengan 4 baris.

Pada aplikasinya tidak semua pin pada LCD 20x4 terpakai, melainkan hanya pin-pin yang dihubungkan dengan mikrokontroler yang digunakan saja. Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. *Interface* LCD merupakan sebuah paralel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 *nibble* data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa *clock* EN setiap *nibble*-nya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroler mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus men-*set* EN ke kondisi *high* “1” dan kemudian mengatur dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus.

Saat jalur lainnya sudah siap, EN harus diatur ke “0” dan tunggu beberapa saat (tergantung pada *datasheet* LCD), dan *set* EN kembali ke *high* “1”. Ketika jalur RS berada dalam kondisi *low* “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi *high* atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan dilayar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus di-*set* ke “1”. Jalur kontrol R/W harus berada dalam kondisi *low* (0) saat informasi pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi *high* “1”, maka program akan melakukan *query* (pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu *Get LCD* status (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu di-*set* ke

“0”. Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7. Mengirim data secara paralel baik 4-bit atau 8-bit merupakan 2 mode operasi primer.

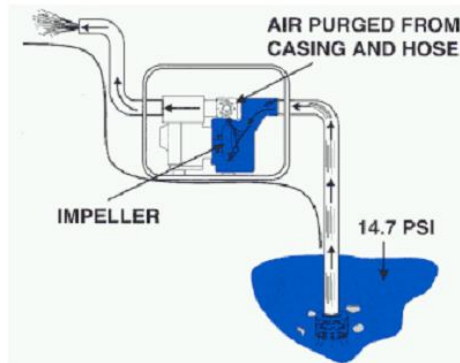
Mode 8-bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 pin I/O (3 pin untuk kontrol, 8 pin untuk data). Sedangkan mode 4 bit minimal hanya membutuhkan 7-bit (3 pin untuk kontrol, 4 pin untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroler dan LCD. Jika bit ini di-*set* ( $RS = 1$ ), maka *byte* pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini di-*reset* ( $RS = 0$ ), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca.

## 2.16 Pompa Air[11]

Pompa air adalah suatu peralatan mekanik yang digerakkan oleh tenaga mesin yang digunakan untuk memindahkan cairan (fluida) dari suatu tempat ke tempat lain, dimana cairan tersebut hanya mengalir apabila terdapat perbedaan tekanan. Pompa juga dapat diartikan sebagai alat untuk memindahkan energi dari pemutar atau penggerak ke cairan ke bejana yang bertekanan yang lebih tinggi. Selain dapat memindahkan cairan pompa juga berfungsi untuk meningkatkan kecepatan, tekanan dan ketinggian cairan.

Pompa hidroponik atau akuarium ini bekerja menggunakan sistem pompa sentrifugal. Pompa sentrifugal atau *centrifugal pumps* adalah pompa yang mempunyai elemen utama yakni berupa motor penggerak dengan sudu *impeller* yang berputar dengan kecepatan tinggi. Prinsip kerjanya yakni mengubah energi mekanis alat penggerak menjadi energi kinetis fluida (kecepatan) kemudian fluida di arahkan ke saluran buang dengan memakai tekanan (energi kinetis sebagian fluida diubah menjadi energi tekanan) dengan menggunakan *impeller* yang berputar di dalam *casing*. *Casing* tersebut dihubungkan dengan saluran hisap (*suction*) dan saluran tekan (*discharge*), untuk menjaga agar di dalam *casing* selalu terisi dengan cairan sehingga saluran hisap harus dilengkapi dengan katup kaki (*foot valve*). Daya dari motor diberikan kepada poros pompa untuk memutar *impeller* yang terpasang pada poros tersebut. Zat cair yang ada didalam *impeller* akan ikut berputar karena dorongan sudu-sudu.

Karena timbul gaya sentrifugal maka zat cair mengalir dari tengah *impeller* akan keluar melalui saluran diantara sudu-sudu dan meninggalkan *impeller* dengan kecepatan tinggi. Zat cair yang keluar dari *impeller* dengan kecepatan tinggi ini kemudian akan keluar melalui saluran yang penampangnya makin membesar (*volute/difuser*) sehingga terjadi perubahan dari *head* kecepatan menjadi *head* tekanan. Oleh sebab itu zat cair yang keluar dari *flens* pompa memiliki *head* total yang lebih besar. Pada Gambar 2.20 merupakan ilustrasi dari prinsip kerja pompa sentrifugal.



**Gambar 2.20** Prinsip Kerja Pompa Sentrifugal

## 2.17 Bluetooth [12]

*Bluetooth* merupakan spesifikasi industri untuk jaringan kawasan pribadi (*personal area networks* atau PAN) tanpa kabel. *Bluetooth* menghubungkan dan dapat dipakai untuk melakukan tukar-menukar informasi di antara peralatan-peralatan. Spesifikasi dari peralatan *bluetooth* ini dikembangkan dan didistribusikan oleh kelompok *Bluetooth Special Interest Group*.

Teknologi *bluetooth* dapat beroperasi dalam pita frekuensi 2,4 GHz dengan menggunakan sebuah *frequency hopping traceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real time* antara *host-host bluetooth* dengan jarak terbatas. Kelebihan yang dimiliki *bluetooth* antara lain, dapat menembus dinding dan berbagai benda lain, tidak memerlukan kabel ataupun kawat, dapat mensinkronisasi basis data dari *handphone* ke komputer, serta dapat digunakan sebagai perantara modem. Sedangkan kekurangan yang dimiliki diantaranya, apabila dalam suatu ruangan terlalu banyak

koneksi *bluetooth* yang digunakan, akan menyulitkan pengguna untuk menemukan penerima yang diharapkan. Kelemahan dari teknologi ini adalah jangkauannya yang pendek dan kemampuan transfer data yang rendah.

Teknologi *bluetooth* menyediakan layanan komunikasi *point to point* maupun *point to multipoint*. Produk *bluetooth* dapat berupa PC card atau USB adapter yang dimasukkan ke dalam perangkat. Sedangkan perangkat yang dapat dikombinasikan dengan *bluetooth* diantaranya, *handphone*, kamera, *personal computer* (PC), printer, headset, dan lain-lain.

Jangkauan layanan *bluetooth* bervariasi, yakni antara 1 meter, 10 meter dan 100 meter. Besar jangkauan layanan ini tergantung pada daya *transmitter*, semakin besar daya maka semakin besar pula jauh jarak jangkauan transmisinya. Berdasarkan jangkauannya, *bluetooth* dapat dikategorikan ke dalam tiga kelas yang terdapat pada Tabel 2.3 berikut :

**Tabel 2.3** Kelas *Bluetooth*

Kelas	Daya (mW)	Jangkauan (meter)
1	100	100
2	10	10
3	1	1

Topologi jaringan pada teknologi *bluetooth* dibagi menjadi dua yaitu *piconet* dan *scatternet*. Di dalam melakukan komunikasi data, *bluetooth* dibagi menjadi dua bagian yaitu *master* dan *slave*. Komunikasi yang terbangun antara *master* dan *slave* dinamakan *piconet*. *Piconet* ini terdiri dari *point to point* dan *point to multipoint*. Agar dapat berkomunikasi, antara *master* dan *slave* harus memiliki frekuensi yang sama. Sedangkan *scatternet* merupakan beberapa jaringan *piconet* yang saling berhubungan dan berkomunikasi antara yang satu dengan yang lain.

Pada umumnya, untuk menggunakan perangkat *bluetooth* maka dibutuhkan modul *bluetooth*. Dalam pengimplementasiannya, sebuah modul *bluetooth* memiliki pengaturan masing-masing. Salah satu pengaturan yang umum digunakan yaitu pengaturan AT Command. AT Command berfungsi untuk melakukan pengaturan konfigurasi dari modul *bluetooth*. Pengaturan AT Command ini memiliki

beberapa perintah yang dapat digunakan antara lain terdapat pada Tabel 2.4 :

**Tabel 2.4** Perintah AT *Command*

No.	Perintah	Kegunaan
1.	AT	untuk menguji apakah modul telah siap digunakan. Apabila respon dari perintah tersebut tertulis “OK” pada <i>serial monitor</i> maka modul <i>bluetooth</i> siap digunakan.
2.	AT + RESET	untuk melakukan perintah <i>reset</i> modul <i>bluetooth</i>
3.	AT + VERSION	untuk melihat versi dari modul <i>bluetooth</i>
4.	AT + ORGL	untuk mengembalikan ke pengaturan awal ( <i>defaults status</i> ) modul
5.	AT + ADDR?	untuk mengetahui alamat dari <i>bluetooth</i>
6.	AT + NAME?	untuk mengetahui nama <i>bluetooth</i>
7.	AT + PSWD?	mengetahui <i>password bluetooth</i>
8.	AT + UART?	untuk mengetahui serial parameter berupa <i>baud rate</i> , <i>stop bit</i> , dan <i>parity</i> yang sedang digunakan.
9.	AT + UART= <Param>,<Param2>, <Param3>	untuk melakukan pengaturan serial parameter yang akan digunakan. <Param> merupakan <i>baud rate</i> , <Param2> merupakan <i>stop bit</i> , dan <Param3> merupakan <i>parity</i> .

Pada pengaturan serial parameter, terdapat tiga parameter yang digunakan yaitu *baud rate*, *stop bit*, dan *parity*. *Baud rate* merupakan besaran kecepatan pengiriman data oleh *bluetooth* dengan satuan bits/s. *Baud rate* yang digunakan yaitu 2400, 4800, 9600, 38400. *Stop bit* yang dapat digunakan yaitu 1 dan 2. Sedangkan untuk *parity* yaitu 0 (*None*), 1 (*Odd*), dan 2 (*Even*).

## 2.18 Sistem Operasi *Android* [13][14]

Sistem operasi *Android* merupakan sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Pada awalnya, sistem operasi *Android* ini dikembangkan oleh *Android, Inc.*, dan kemudian dibeli oleh Google dan mengalami banyak perkembangan sampai sekarang. Antarmuka pengguna *Android* menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk dan lain lain. Saat ini, perkembangan sistem operasi *Android* telah mencapai versi ke 7.0. Tabel perkembangan versi *Android* terdapat pada Tabel 2.5.

Sistem operasi *Android* tersusun atas element – element seperti *application layer*, *application framework*, *libraries*, dan Linux *kernel*. Kernel linux adalah layanan inti (termasuk didalamnya driver hardware, manajemen proses dan memori, keamanan, *network*, dan manajemen daya) yang dikerjakan oleh kernel Linux 2.6. kernel juga menyediakan lapisan abstraksi (*abstraction layer*) antara hardware dan elemen lainnya. Pustaka (*library*) berjalan pada bagian atas dari kernel, termasuk didalamnya berbagai macam pustaka inti seperti pustaka (*library*) media yang digunakan untuk memainkan audio dan video, *surface manager* untuk menyediakan manajemen *display*, pustaka grafik (*Graphics libraries*) yang didalamnya *SGL* dan *openGL* untuk grafis 2D dan 3D, *SQLite* untuk layanan basis data, *SSL* dan *Webkit* untuk *web browser* dan keamanan internet (*Internet Security*) terintegrasi.

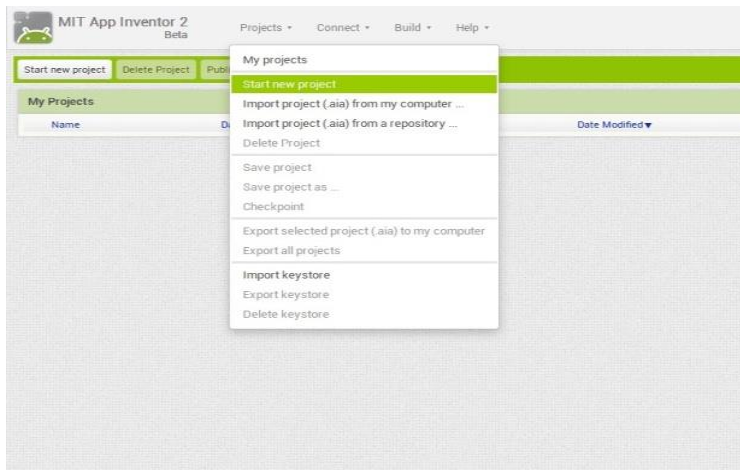
**Tabel 2.5** Tingkatan Versi *Android*

Versi	Nama Kode	Tahun Rilis
1.5	<i>Cupcake</i>	2009
1.6	<i>Donut</i>	2009
2.0-2.1	<i>Éclair</i>	2009
2.2	<i>Froyo</i>	2010
2.3-2.3.2	<i>Gingerbread</i>	2010
2.3.3-2.3.7	<i>Gingerbread</i>	2011
3.1	<i>Honeycomb</i>	2011
3.2	<i>Honeycomb</i>	2011
Versi	Nama Kode	Tahun Rilis
4.0.3-4.0.4	<i>Ice Cream Sandwich</i>	2011
4.1.x	<i>Jelly Bean</i>	2012

<b>Versi</b>	<b>Nama Kode</b>	<b>Tahun Rilis</b>
4.2.x	<i>Jelly Bean</i>	2012
4.3.x	<i>Jelly Bean</i>	2013
4.4.x	<i>KitKat</i>	2013
5.x	<i>Lollipop</i>	2014
6.0	<i>Marshmallow</i>	2015
7.0	<i>Nougat</i>	2016

### 2.19 App Inventor[14]

App Inventor merupakan aplikasi web sumber terbuka yang dikelola oleh *Maschusetts Institute of Technology* (MIT). App Inventor memungkinkan pengguna baru untuk memprogram komputer untuk menciptakan aplikasi perangkat lunak bagi sistem operasi *Android*. Dengan menggunakan antarmuka grafis, App Inventor memungkinkan pengguna untuk melakukan *drag and drop* objek visual berupa blok-blok untuk menciptakan aplikasi yang dapat dijalankan dengan perangkat *Android*. Dalam menjalankan App Inventor, pengguna cukup *login email* dan dapat membuat *project* baru. Dalam menjalankannya harus terhubung dengan jaringan internet karena program ini hanya dapat digunakan secara *online*. Terdapat dua tampilan utama pada App Inventor, yaitu *Design View* dan *Blocks View*. Pada *Design View* memungkinkan pengguna untuk membuat tampilan *user interface* untuk aplikasi *Android*. Sedangkan pada *Blocks View*, memungkinkan pengguna untuk membuat program dari App Inventor. Pada Gambar 2.21 berikut merupakan tampilan awal dari App Inventor.



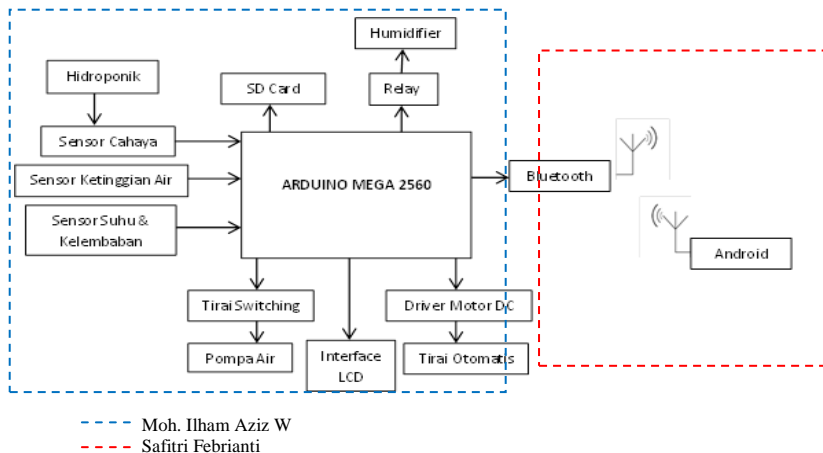
**Gambar 2.21** Tampilan Awal App Inventor



### BAB III

## PERANCANGAN *HARDWARE* DAN *SOFTWARE*

Pada bab ini membahas tentang tahapan yang dilakukan terhadap perancangan dan pembuatan Tugas Akhir yang berjudul Sistem Otomasi Penyemaian Benih Sayuran Hidroponik pada Kebun Sayur Surabaya. Dimana akan dibagi menjadi dua bagian untuk Moh. Ilham Aziz W. bertugas untuk pembuatan rangkaian *shield* Arduino Mega 2560, rangkaian *solid state relay* (SSR) menggunakan TRIAC, rangkaian *power supply*, pembuatan rangkaian pengondisi sinyal sensor HSM 20-G, perancangan sensor *water level* Funduino, perancangan sensor cahaya, perancangan rangkaian RTC (*Real Time Clock*), perancangan *driver* motor L298N, merancang *relay* sebagai *switching*, perancangan modul *SD card*. Sedangkan untuk Safitri Febrianti yaitu bertugas untuk melakukan pengaturan modul *blue-tooth* HC-05, merancang aplikasi *Android*, komunikasi data *Android* dengan alat, *monitoring* temperatur dan kelembaban serta kontrol aktuatur menggunakan *Android*.



**Gambar 3.1** Diagram Fungsional Sistem

Berikut penjelasan mengenai blok diagram pada Gambar 3.1 yaitu Arduino Mega 2560 sebagai mikrokontroler. Sensor air mendeteksi ketinggian air pada objek yaitu benih sayuran hidro-

ponik, jika ketinggian air melebihi *setting point*, maka mikrokontroler akan melakukan penanganan dengan mengaktifkan rangkaian TRIAC yang berfungsi untuk menonaktifkan pompa air. Begitu pula dengan sensor temperatur dan kelembaban yang akan mendeteksi kondisi objek hidroponik, apabila tidak sesuai *setting point* maka mikrokontroler akan mengaktifkan rangkaian *switching relay* yang berfungsi untuk mengaktifkan *ultrasonic humidifier*. Saat *ultrasonic humidifier* aktif maka akan mengeluarkan uap air yang diarahkan ke objek. Fungsi dari SD Card ini adalah untuk menyimpan nilai *setting point* dari setiap sayuran dan juga berfungsi untuk *datalogger*. *Interface* yang digunakan pada alat ini menggunakan LCD 20x4, kemudian untuk mengatur intensitas cahaya yang dibutuhkan disini menggunakan sensor cahaya LDR yang nantinya akan mengaktifkan *driver* motor DC untuk membuka atau menutup tirai otomatis. Untuk memudahkan pengguna dalam proses penyemaian sayuran, pengguna juga dapat melihat kondisi sayuran tersebut melalui *Android*. Modul *bluetooth* HC-05 ini digunakan untuk komunikasi data antar Arduino dengan *Android*.

### 3.1 Perancangan *Hardware*

Pada perancangan *hardware* dibagi menjadi beberapa sub bab yang akan dijelaskan per sub bab nya, antara lain :

1. Rangkaian *Shield* Arduino Mega 2560
2. Rangkaian *Solid State Relay* (SSR) menggunakan TRIAC
3. Rangkaian *Power Supply*
4. Rangkaian Pengondisi Sinyal Sensor HSM 20G
5. Perancangan Sensor *Water Level* Funduino
6. Perancangan Sensor Cahaya LDR
7. Rangkaian RTC (*Real Time Clock*) DS1307
8. Perancangan *Driver* Motor L298N
9. Perancangan *Hardware Plant*
10. Perancangan *Relay* sebagai *Switching*
11. Perancangan Modul SD Card
12. Pengaturan Modul *Bluetooth* HC-05

#### 3.1.1 Perancangan Rangkaian *Shield* Arduino Mega 2560

Pada perancangan rangkaian kontroler ini menggunakan Arduino Mega 2560 untuk menerima masukan dari sensor ketinggian

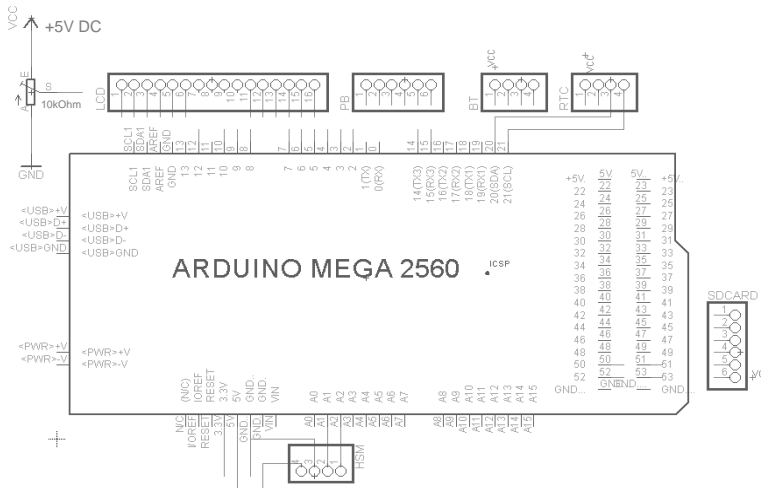
air, temperatur, dan kelembaban, menerima data waktu dari rangkaian RTC, menulis data ke SD Card, serta menjalankan program untuk komunikasi alat dengan Android via *bluetooth*. Pada rangkaian ini ditambahkan sebuah *shield* berupa rangkaian LCD dengan *pushbutton*. Pada *shield* ini terdapat *port* analog dan *port* digital yang dihubungkan dengan *port* pada arduino. Arduino sebagai mikrokontroler diprogram agar dapat mengontrol kadar ketinggian air, kadar temperatur dan kelembaban di dalam *box plant*, menyalakan dan mengaktifkan pompa air dan *humidifier*, mengatur pengiriman data via *bluetooth*, dan menyimpan data dalam SD Card. Berikut ini pada Tabel 3.1 merupakan tabel untuk penggunaan pin pada *shield* Arduino yang akan dibuat :

**Tabel 3.1** Mapping Pin Analog/Digital yang digunakan.

No.	Modul	Pin Analog/Digital
1.	RTC	VCC : 5V GND SDA : Digital 20 SCL : Digital 21
2.	HSM-20G	VCC : 5V GND TEMPERATURE : Analog 1 HUMIDITY : Analog 2
3.	SD CARD	VCC :3,3-5V GND DO (MISO) : Digital 50 DI (MOSI) : Digital 51 CLK : Digital 52 CS : Digital 53
4.	WATER SENSOR	VCC : 5V GND DATA(S) : Analog 0
5.	BLUETOOTH	VCC : 5V GND RX : Digital 18 (TX1) TX : Digital 19 (RX1)
6.	PUSH BUTTON	UP : Digital 2 DOWN : Digital 3 SELECT : Digital 4 BACK : Digital 12

No.	Modul	Pin Analog/Digital
7.	<i>LIQUID CRYSTAL DISPLAY 20X4</i>	VSS : GND VDD : VCC 5V VO : Tersambung dengan potensiometer RS : <i>Digital</i> 10 RW : GND E : <i>Digital</i> 9 D4 : <i>Digital</i> 36 D5 : <i>Digital</i> 38 D6 : <i>Digital</i> 6 D4 : <i>Digital</i> 5 A (Anoda) : VCC K (Katoda) : GND
8.	<i>DRIVER MOTOR L298N</i>	VCC : 6V dari <i>Power Supply</i> GND EN(PWM) : <i>Digital</i> 11 IN1 : <i>Digital</i> 33 IN2 : <i>Digital</i> 34
9.	<i>ULTRASONIC HUMIDIFIER</i>	VCC : 24 V GND ; <i>Switching</i> dari <i>Relay Digital</i> 39
10.	POMPA AIR	VAC 220V <i>Switching</i> dari Triac <i>Digital</i> 37

Berikut adalah rangkaian modul *shield* Arduino Mega 2560 yang diperlukan untuk alat ini. Modul *shield* ini digunakan untuk mempermudah koneksi antar modul-modul yang digunakan dengan Arduino, selain itu dengan adanya modul ini dapat mengurangi jumlah *wiring* pada alat ini. Untuk lebih jelasnya terdapat pada Gambar 3.2 yang merupakan skematik dari rangkaian modul *shield* Arduino Mega 2560.



**Gambar 3.2** Rangkaian Modul Shield Arduino Mega 2560

### 3.1.2 Perancangan Rangkaian Solid State Relay (SSR) Menggunakan TRIAC

Rangkaian ini terdiri dari *optocoupler* MOC3041, TRIAC BT138, transistor BC337, LED, dan resistor 1k $\Omega$ , 220 $\Omega$ , 330 $\Omega$ . Rangkaian ini bertujuan sebagai saklar dari pompa air. *Optocoupler* MOC3041 berfungsi sebagai pengaman kendali antara sumber AC dan DC, dikarenakan rangkaian ini berhubungan langsung dengan sumber AC jika sewaktu-waktu terdapat kesalahan. Di dalam *optocoupler* MOC3041 terdapat *zero cross circuit* yang berfungsi untuk menjaga *optocoupler* pada saat VAC mencapai 0 V, serta melindungi adanya arus yang besar saat tegangan AC berada pada puncak tegangan. Resistor R1 digunakan untuk membatasi nilai arus yang masuk ke LED tidak melebihi dari yang dibutuhkan LED, sesuai hukum Ohm, ketika keluaran pin digital arduino bernilai *HIGH* akan menghasilkan tegangan sebesar 4,9V. LED mempunyai arus kerja maksimum sebesar 20mA. Jadi, kurang lebih dari perhitungan disini didapatkan 4,9mA agar LED dapat menyala dengan memakai resistor sebesar 1K $\Omega$ . Pada kaki basis transistor BC337 diberi resistor karena arus yang akan mengalir dari basis harus lebih kecil daripada arus yang mengalir dari kolektor ke emitor, oleh sebab itu

maka ada baiknya jika pin basis dipasang sebuah resistor. Nilai resistor R2 dari :

$$R_b = (V_{in} - V_{BE}) / I_B \dots\dots\dots(3.1)$$

$$I_B = I_c / \beta \dots\dots\dots(3.2)$$

Nilai  $I_c=500$  dan  $\beta=100$  dari *datasheet* transistor BC337. Berikut nilai perhitungan nilai resistor yang dibutuhkan pada basis arduino :

$$I_B=500/100$$

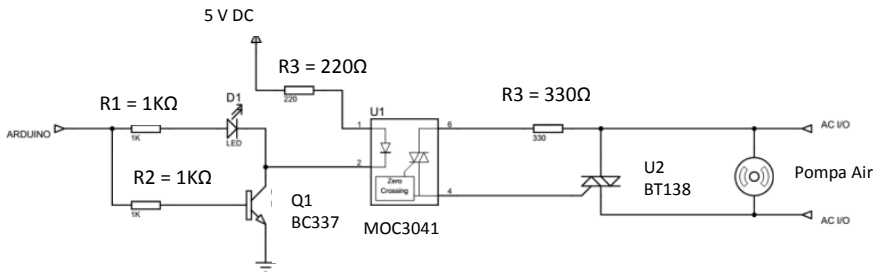
$$I_B=5mA$$

$$R_B = (4,9 - 0,7) / 0,005$$

$$R_B = 4,2 / 0,005$$

$$R_B = 840\Omega$$

Didapatkan nilai RB kurang lebih sebesar  $840\Omega$  untuk pin basis transistor BC337.



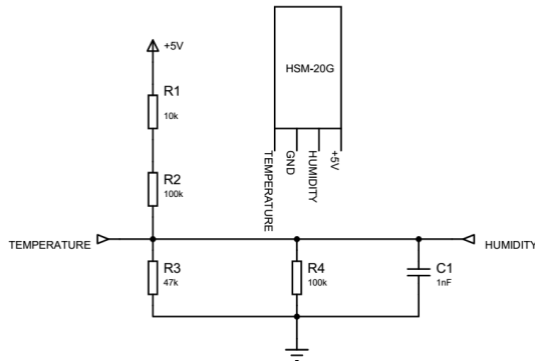
**Gambar 3.3** Skematik Rangkaian SSR Menggunakan TRIAC

TRIAC BT138 merupakan salah satu jenis *thyristor*. Komponen ini berfungsi sebagai saklar (*switching*) dengan sumber tegangan AC. Transistor BC337 merupakan jenis transistor NPN, transistor ini tersambung dengan pin digital Arduino, transistor BC337 ini digunakan untuk menonaktifkan rangkaian *switching* TRIAC. Transistor BC 337 jika aktif akan memicu rangkaian dalam pada *optocoupler* dan mengaktifkan fungsi *switching* pada TRIAC BT 138.



### 3.1.4 Perancangan Rangkaian Pengondisi Sinyal Sensor HSM 20-G

Sensor temperatur dan kelembaban bertipe HSM 20-G membutuhkan rangkaian pengondisi sinyal dengan komponen resistor dan kapasitor. Skematik rangkaian pengondisi sinyal HSM 20-G terdapat pada Gambar 3.5.



**Gambar 3.5** Skematik Rangkaian Pengondisi Sinyal Sensor HSM 20-G

Pada Gambar 3.5 merupakan rangkaian pengondisi sinyal dari sensor temperatur dan kelembaban HSM-20G. Seperti yang telah dijelaskan sebelumnya bahwa sensor ini memiliki 2 keluaran yakni temperatur dan kelembaban. Untuk keluaran temperatur yang berupa resistansi, diperlukan rangkaian pembagi tegangan dengan menggunakan resistor  $47k\Omega$  dan  $110k\Omega$  sehingga dapat diukur keluaran tegangannya. Resistor  $47k\Omega$  dipilih karena dalam penelitian ini menggunakan ketetapan temperatur ruang  $25^{\circ}\text{C}$  (menurut *datasheet* sensor yang sudah diujikan). Sedangkan untuk keluaran kelembaban, digunakan rangkaian paralel antara resistor  $100k\Omega$  dengan kapasitor  $47\mu\text{F}$  sehingga didapatkan data keluaran sensor berupa tegangan.

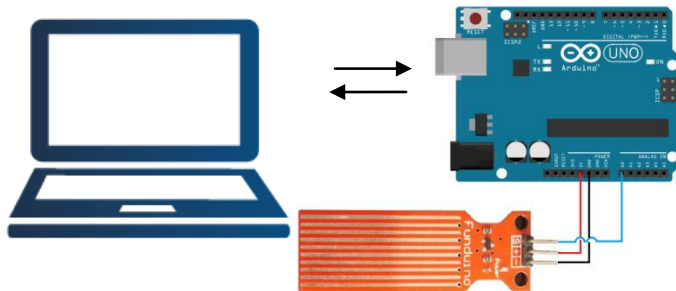
### 3.1.5 Perancangan Sensor Water Level Funduino

Sensor ketinggian air merupakan sensor untuk mengukur ketinggian air dan deteksi air. Sensor ini dapat mengukur tinggi air sebesar 4 cm. Setiap baris metal dari sensor ini mempunyai resistansi yang berbeda, semakin dalam sensor tersebut dimasukkan air semakin besar pula nilai resistansinya. Sensor *water level* keluaran



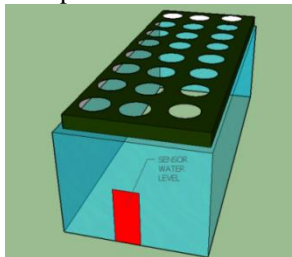
*Funduino* ini mempunyai tingkat akurasi ketepatan yang cukup bagus.

Sensor *water level* disini ditempatkan pada kotak bagian dalam *starter kit* pada plant tanaman sayuran. Sensor ini digunakan untuk mengatur kadar air yang dibutuhkan untuk tanaman. Gambar 3.6 adalah konfigurasi sensor *water level* dengan Arduino :



**Gambar 3.6** Konfigurasi Sensor Funduino

Terdapat tiga pin yang digunakan untuk sensor ini pin VCC tersambung dengan *input* 5V pada pin Arduino, pin GND terdambung dengan pin GND pada Arduino dan pin S tersambung pada pin *Analog* 0 pada pin Arduino. Pada Gambar 3.7 dibawah merupakan peletakan sensor pada *starter kit*.



**Gambar 3.7** Peletakan Sensor Funduino pada *Box Starter Kit*

### 3.1.6 Perancangan Sensor Cahaya LDR

Sensor cahaya LDR dalam tugas akhir ini menggunakan modul sensor cahaya yang sudah dilengkapi dengan rangkaian tambahan guna pengondisian sinyal sensor ini agar keluaran dari sensor tersebut mempunyai nilai yang cukup akurat.

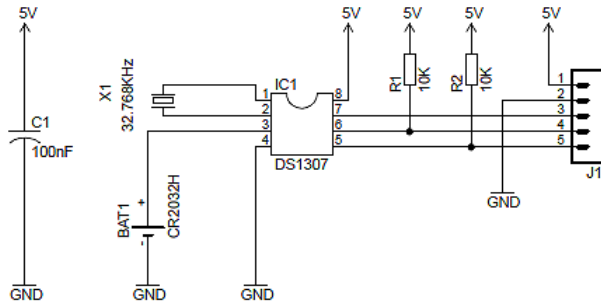
Tabel 3.2 berikut adalah spesifikasi dari sensor modul LDR yang digunakan :

**Tabel 3.2** Konfigurasi Pin Modul Sensor LDR dengan Arduino

Pin	Deskripsi	Fungsi
VCC	+3,3V-5V	Disambungkan ke 3,3V-5V Arduino
GND	0V	Disambungkan ke <i>Ground</i> Arduino
D0	<i>Digital Output</i>	<ol style="list-style-type: none"> <li>1. Sinyal <i>Output</i> : <i>HIGH</i> Intensitas lampu di sekitar ruangan sampai pre-set level. (Diatur dengan pengatur sensitivitas pada potensiometer) LED D0 : ON</li> <li>2. Sinyal <i>Output</i> : <i>LOW</i> Intensitas lampu di sekitar ruangan tidak sampai pre-set level. (Diatur dengan pengatur sensitivitas pada potensiometer) ; LED D0 : OFF</li> </ol>
POT	Potensiometer	<ol style="list-style-type: none"> <li>1. Jika diputar searah jarum jam meningkatkan kepekaan terhadap intensitas cahaya.</li> <li>2. Jika berlawanan jarum jam untuk mengurangi kepekaan terhadap intensitas cahaya.</li> </ol>
A0	<i>Analog Output</i>	Disambungkan dengan pin Analog 0 Arduino

### 3.1.7 Perancangan RTC(*Real Time Clock*) DS1307

Pemakaian RTC dalam alat ini digunakan untuk menampilkan tanggal dan waktu pada LCD serta yang paling utama difungsikan sebagai tanggal dan waktu pada *datalogger*. Gambar 3.8 berikut adalah rangkaian modul RTC DS1307.

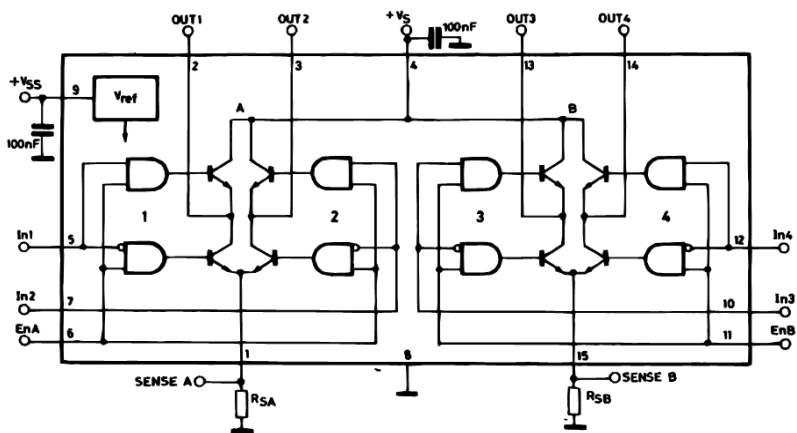


**Gambar 3.8** Rangkaian Modul RTC DS1307

### 3.1.8 Perancangan *Driver* Motor L298N

*Driver* motor L298N merupakan *driver* motor yang paling populer digunakan untuk mengontrol atau mengendalikan kecepatan dan arah pergerakan motor terutama untuk motor DC. Untuk IC utama yaitu IC L298N merupakan sebuah IC tipe *H-bridge* yang mampu mengendalikan beban-beban induktif seperti *relay*, solenoid, motor DC dan motor *stepper*. Pada ICL298N terdiri dari transistor-transistor logik (TTL) dengan gerbang nand yang berfungsi untuk memudahkan dalam menentukan arah putaran suatu motor DC maupun motor *stepper*. *Driver* motor L298N juga paling populer digunakan untuk mengontrol kecepatan dan arah pergerakan motor terutama pada robot *line follower / line tracer*. Kelebihan dari *driver* motor L298N ini adalah cukup presisi dan mudah dalam mengontrol motor. Pada prinsipnya rangkaian *driver* motor L298N ini dapat mengatur tegangan dan arus sehingga kecepatan dan arah motor dapat diatur.

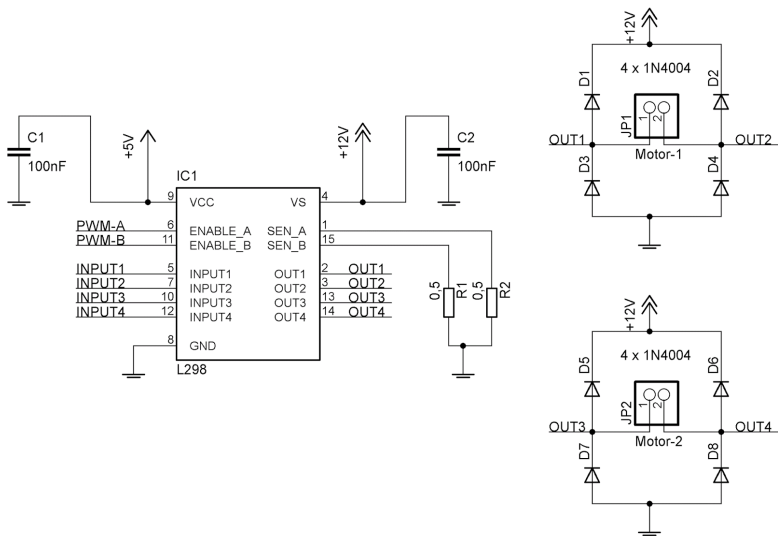
Perancangan *driver* motor L298N ini digunakan untuk mengatur kecepatan motor DC serta mengatur posisi dari motor DC tersebut. Motor DC disini digunakan untuk membuka dan menutup atap secara otomatis. Prinsip kerja yang digunakan *driver* ini menggunakan *H-Bridge*.



**Gambar 3.9** Rangkaian *H-Bridge* pada Driver Motor L298N

Pada Gambar 3.9 tampak bahwa pada masing-masing kaki basis transistor *H-bridge* dihubungkan dengan sebuah gerbang logika AND yang salah satu kaki *input*-nya digabung dan dihubungkan dengan kaki In1 (*Input 1*) dan In2 (*Input 2*). Kemudian *input* salah satu gerbang AND (yaitu gerbang AND bagian bawah) diberi *inverter* (pembalik kondisi) yang berfungsi untuk pembalik sinyal. Selanjutnya (masih pada *H-bridge* sebelah kiri pada IC L298), kaki *input* yang kedua pada keempat gerbang AND dihubungkan dengan kaki EnA (*Enable A*). Kaki EnA berfungsi untuk mengaktifkan atau menonaktifkan rangkaian *H-bridge* pada IC L298.

IC L298 dapat bekerja dengan tegangan catu hingga 46 V DC dan memiliki arus (DC) kerja maksimal hingga 4 Ampere. Dengan spesifikasi tersebut, IC L298 sudah dapat digunakan dalam mengendalikan putaran motor DCMP dengan arus kerja hingga 4 Ampere. IC L298 memiliki 15 kaki yang memiliki fungsi tersendiri.



**Gambar 3.10** Rangkaian *Driver* Motor L298N

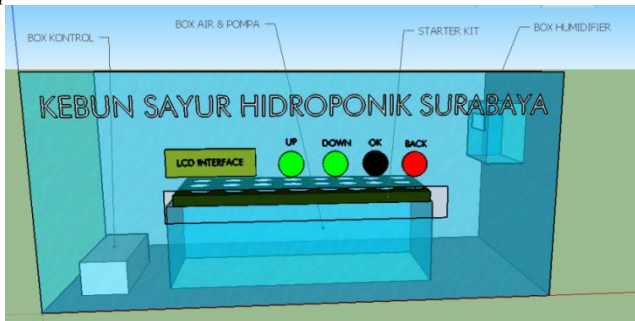
Gambar 3.10 di atas merupakan contoh skematik rangkaian *Driver* motor DCMP menggunakan IC L298. Terdapat 4 kaki *input*, yaitu kaki *input1* hingga kaki *input4* (pada pin 5, 7, 10, dan 12 IC L298). Kaki *input* digunakan untuk menentukan arah putar motor DCMP, apakah berputar CW atau berputar CCW. Perlu diketahui bahwa, keempat kaki *input* pada IC L298 adalah terbagi menjadi 2 pasangan, yaitu kaki *input1* dan kaki *input2* digunakan untuk menentukan arah putar motor DCMP yang dihubungkan pada kaki *output1* dan kaki *output2*. Sementara itu, kaki *input3* dan kaki *input4* digunakan untuk menentukan arah putar motor DCMP yang dihubungkan pada kaki *output3* dan kaki *output4*.

Kemudian pada gambar tersebut juga terdapat kaki *Enable\_A* dan kaki *Enable\_B*. Kaki *Enable* tersebut harus diberi logika *high* (1) ketika hendak mengaktifkan rangkaian *H-bridge* yang ada di dalam IC L298. Sedangkan jika hendak menon-aktifkan rangkaian *H-bridge* yang ada di dalam IC L298, maka kaki *Enable* harus diberi logika *low* (0). Seperti halnya kaki *input*, kaki *Enable* pada IC L298 adalah terbagi menjadi 2, yaitu *Enable\_A* untuk mengaktifkan/menon-aktifkan motor DCMP yang dihubungkan pada

kaki *output1* dan kaki *output2*. Sedangkan *Enable\_B* untuk mengaktifkan/menon-aktifkan motor DCMP yang dihubungkan pada kaki *output3* dan kaki *output4*.

### 3.1.9 Perancangan *Hardware Plant*

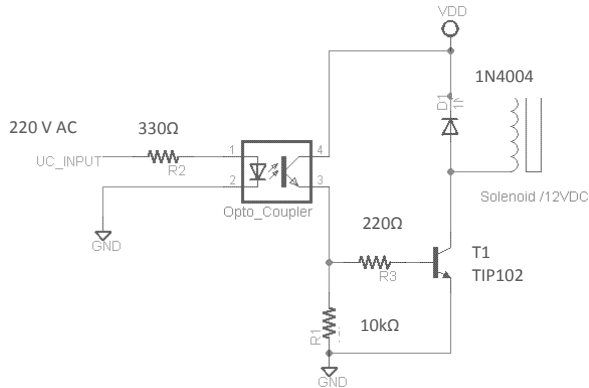
Perancangan *hardware plant* bertujuan untuk menempatkan tanaman hidroponik beserta rangkaian elektrik yang digunakan, sehingga dibutuhkan *box plant*. Terdapat beberapa bagian dari box ini, diantaranya, bak air, *starter kit* sebagai *plant* dari tumbuhan, box untuk tempat *humidifier*, *box control* untuk tempat beberapa rangkaian elektronik, serta pada bagian depan terdapat LCD 20x4 sebagai tampilan *interface* dan *push button* untuk mengoperasikan alat ini. Pada Gambar 3.11 berikut ini merupakan sketsa perancangan *box plant*:



**Gambar 3.11** Perancangan Desain *Hardware Plant*

### 3.1.10 Perancangan *Relay* sebagai *Switching*

Pada rangkaian *driver relay* ini digunakan *relay* 12 V, IC *optocoupler* PC817, transistor C829, resistor 330 $\Omega$ , 100 $\Omega$ , dan 1K $\Omega$ . IC *optocoupler* (PC817) yang berfungsi sebagai proteksi arus, agar arus yang digunakan untuk memicu transistor C829 tidak dapat bercampur atau dengan kata lain merusak komponen Arduino Mega 2560. Sedangkan transistor C829 digunakan untuk proses *switching* alasan menggunakan transistor yang memiliki tegangan kerja lebih besar seperti BD139 misalnya. *Driver relay* pada alat ini digunakan untuk *switching ultrasonic humidifier*. Pada Gambar 3.12 berikut ini merupakan skematik perancangan *driver relay* :



**Gambar 3.12** Perancangan *Driver Relay*

### 3.1.11 Perancangan Modul *SD Card*

Modul *SD Card* adalah sebuah modul yang berfungsi untuk membaca dan menulis data ke atau dari *SD Card*. Modul ini memiliki *interfacing* menggunakan komunikasi SPI. Tegangan kerja dari modul ini dapat menggunakan level tegangan 3,3 atau 5 V DC. Modul ini digunakan untuk membuat piranti yang membutuhkan suatu penyimpanan bersifat *non-volatile* yang berarti data akan tetap tersimpan walaupun tidak mendapatkan *supply* tegangan. Modul ini memiliki 8 buah pin, diantaranya: VCC (3,3V DC), GND, CE, CSN, MOSI, MISO, SCK, dan IRQ. Namun pin yang digunakan hanya 6 yaitu terdapat pada Tabel 3.3 :

**Tabel 3.3** *Mapping Pin SD Card dengan Modul Arduino*

VCC	3,3-5 VDC
GND	GND
CS/CSN	Digital I/O 53
DO (MISO)	Digital I/O 50
DI (MOSI)	Digital I/O 51
CLK	Digital I/O 52

### 3.1.12 Pengaturan Modul *Bluetooth HC-05*

Modul *bluetooth* HC-05 merupakan modul komunikasi nirka-bel pada frekuensi 2,4 GHz dengan pilihan koneksi sebagai *master* maupun *slave*. Modul ini memiliki enam pin diantaranya :

- 1) EN berfungsi untuk mengaktifkan mode AT *Command Setup* pada modul HC-05. Pin ini terhubung dengan *button* dan apabila *button* pada pin ini ditekan dan ditahan maka modul akan mengaktifkan mode AT *Command Setup*. Secara *default*, modul ini aktif dalam mode Data.
- 2) VCC merupakan pin yang berfungsi sebagai *input* tegangan.
- 3) GND merupakan pin yang berfungsi sebagai *ground*.
- 4) TXD adalah pin yang berfungsi untuk mengirimkan data dari modul ke perangkat lain.
- 5) RXD berfungsi untuk menerima data yang dikirim ke modul.
- 6) STATE berfungsi untuk memberikan informasi apakah modul terhubung dengan perangkat lain atau tidak

Meskipun terdapat enam pin, namun *interface* yang digunakan adalah serial VCC, GND, TXD dan RXD. Selain itu, modul ini dilengkapi dengan LED berfungsi sebagai indikator koneksi *bluetooth*. Pada Tabel 3.4 merupakan tabel spesifikasi dan pada Tabel 3.5 merupakan tabel *default status* (status awal) dari modul *bluetooth* HC-05 :

**Tabel 3.4** Spesifikasi Modul *Bluetooth* HC-05

Protokol <i>Bluetooth</i>	<i>Bluetooth Specification v2,0+EDR</i>
Frekuensi	2,4 GHz ISM band
Modulasi	GFSK ( <i>Gaussian Frequency Shift Keying</i> )
<i>Emission Power</i>	?4dBm, Class 2
Sensitivitas	?-84dBm at 0,1 % BER
Kecepatan	-Asinkron : 2,1 Mbps (Max) / 160kbps -Sinkron : 1 Mbps / 1 Mbps
<i>Profil</i>	<i>Bluetooth Serial Port</i>
Tegangan	+3,3 V DC
Arus	-Saat <i>paired</i> (terhubung) : 10 mA -Saat <i>unpaired</i> (tidak terhubung) : 30 mA

**Tabel 3.5** *Default Status* Modul *Bluetooth* HC-05

<i>Device type</i>	0
<i>Inquire code</i>	0x009e8b33
<i>Module work mode</i>	<i>Slave mode</i>
<i>Connection mode</i>	<i>connect to the Bluetooth device specified</i>



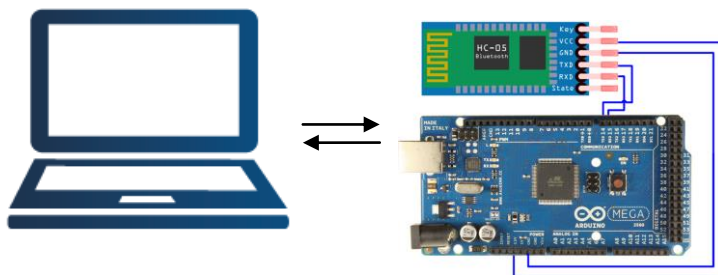
<i>Serial parameter</i>	<i>Baud rate: 38400 bits/s; Stop bit: 1 bit; Parity bit: None</i>
<i>Passkey</i>	“1234”
<i>Device name</i>	“H-C-2010-06-01”
<i>Default Baud Rate</i>	9600, Data Bit 8; Stop Bit 1; Parity 0

Dalam pengaplikasiannya, modul *bluetooth* HC-05 membutuhkan pengaturan terlebih dahulu. Prosedur pengaturan tersebut menggunakan perintah AT atau yang biasa disebut dengan AT *command*.

Seperti yang telah dijelaskan pada bab II, bahwa terdapat pengaturan modul HC-05 menggunakan AT *command*. Sebelum menggunakan perintah AT *command*, modul *bluetooth* HC-05 harus dikoneksikan terlebih dahulu dengan Arduino. Pada Tabel 3.6 merupakan pin *mapping* modul *bluetooth* HC-05 dengan Arduino.

**Tabel 3.6** Pin Mapping Modul Bluetooth HC-05 dengan Arduino saat Perintah AT Command

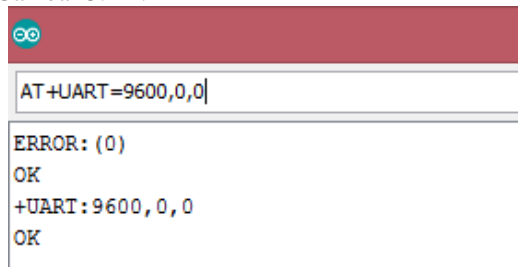
No.	Pin Modul HC-05	Pin Arduino
1.	VCC	VCC
2.	GND	GND
3.	TXD	TXD-3
4.	RXD	RXD-3



**Gambar 3.13** Pengkabelan Modul Bluetooth HC-05 dengan Arduino

Selanjutnya, mengatur perintah AT *command* pada *serial monitor* program Arduino IDE. Untuk mengetahui apakah modul *bluetooth* siap digunakan, maka dilakukan pengujian dengan cara mengetikkan perintah “AT” pada *serial monitor*. Apabila modul

tersebut telah siap digunakan maka akan muncul respon “OK” pada *serial monitor*. Kemudian, mengatur *baud rate* yang akan digunakan. Sebagai contoh, menggunakan *baud rate* 9600, dengan *stop bit* 1 dan *parity* 0. Maka pada *serial monitor*, ketikkan perintah “AT+UART=9600,0,0”. Angka 9600 menunjukkan *baud rate*, angka 0 merupakan *stop bit* yang berarti 1 bit, dan angka 0 menunjukkan nilai *parity* sama dengan 0. Pengaturan tersebut ditunjukkan seperti pada Gambar 3.14 :



**Gambar 3.14** Pengaturan AT Command

Setelah dilakukan pengaturan AT Command, maka proses komunikasi dapat berlangsung. Terdapat perubahan pada pengkabelan untuk proses komunikasi data ini, yakni pada pengkabelan pin TXD dan RXD antara modul *bluetooth* HC-05 dengan arduino. Tabel 3.7 berikut ini merupakan pin *mapping* modul *bluetooth* HC-05 dengan Arduino untuk melakukan komunikasi data.

**Tabel 3.7** Pin Mapping Modul Bluetooth HC-05 dengan Arduino saat melakukan Komunikasi Data

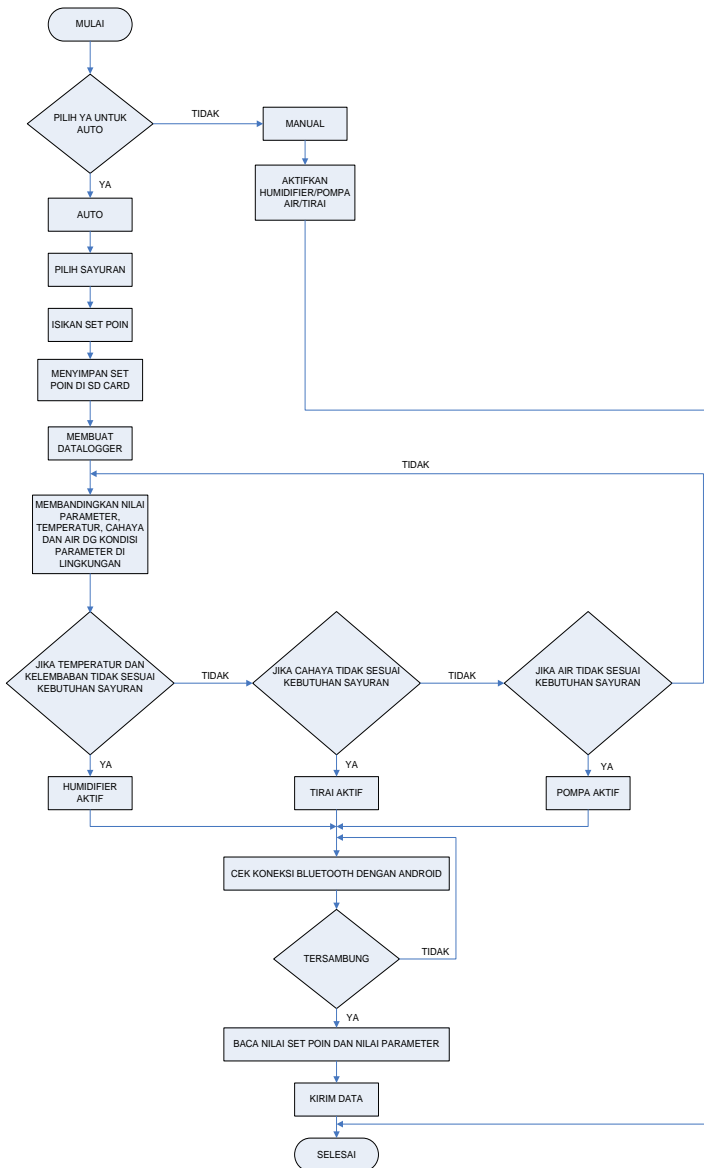
No.	Pin Modul HC-05	Pin Arduino
1.	VCC	VCC
2.	GND	GND
3.	TXD	RXD-3
4.	RXD	TXD-3

### 3.2 Perancangan Software

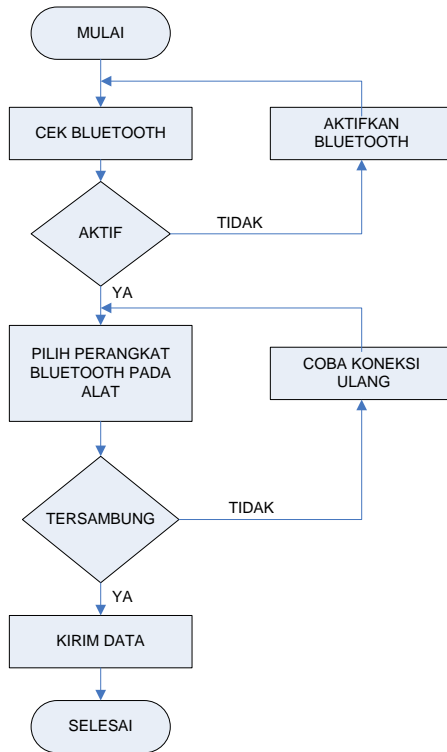
Pada perancangan *software* untuk alat ini dibagi menjadi dua yaitu perancangan *software* untuk alat dan komunikasi data menggunakan *bluetooth* serta perancangan aplikasi *Android* menggunakan App Inventor dan *software* Arduino IDE.

Sebelum merancang program komunikasi data, perlu dilakukan pembuatan program untuk alat. Sistem alat akan berjalan apabila pengguna telah memasukkan *setting point* berupa temperatur dan kelembaban dari sayuran yang akan ditanam. Terdapat dua mode dalam alat ini yaitu mode *Auto* atau *Manual*. Pada pilihan *manual* pengguna dapat melakukan pengontrolan berupa mengaktifkan atau menonaktifkan aktuator-aktuator yang digunakan yakni pompa air dan *humidifier*. Sedangkan pada pilihan *auto*, sistem akan berjalan secara otomatis dalam melakukan penanganan untuk mengaktifkan atau menonaktifkan pompa air dan *humidifier* sesuai dengan perbandingan data parameter terhadap *setting point*. Data-data tersebut selanjutnya akan disimpan pada *SD Card* dan dapat ditampilkan pada *smartphone android* saat pengiriman data melalui *bluetooth*. *Flowchart* keseluruhan sistem pada alat ini terdapat pada Gambar 3.15. Setelah mengisi set poin, parameter tersebut akan disimpan pada *SD Card* dan jika pengguna menginginkan *datalogger*. Sistem akan secara otomatis akan membandingkan nilai *set poin* dengan nilai parameter yang sesuai dengan kondisi lingkungan. Jika temperatur dan kelembaban maka *humidifier* akan aktif, jika kadar cahaya tidak sesuai dengan kebutuhan sayuran maka tirai atau atap akan aktif membuka atau menutup, jika air tidak sesuai dengan kebutuhan sayuran maka pompa aktif. Ketika pengguna ingin mengetahui kondisi dari tanaman yang sedang ditanam, pengguna dapat melihat kondisinya pada *Android*. Komunikasi data yang digunakan dalam transfer data ini menggunakan *Bluetooth*.

Guna melakukan transfer data antara alat dan *smartphone Android*, dibutuhkan komunikasi data menggunakan *bluetooth*. Agar tujuan dari komunikasi data dapat tercapai, maka harus dilakukan pengkoneksian antara perangkat *bluetooth* pada alat dengan perangkat *bluetooth* pada *smartphone*. Yang pertama harus dilakukan yakni mengaktifkan kedua perangkat *bluetooth*, selanjutnya melakukan pencarian perangkat *bluetooth* yang akan dikoneksikan menggunakan aplikasi *Android* pada *smartphone*. Apabila kedua perangkat *bluetooth* telah terhubung, maka data parameter yang ada pada alat akan tampil pada aplikasi, namun jika belum terhubung, maka lakukan pengkoneksian ulang. *Flowchart* komunikasi data *bluetooth* terdapat pada Gambar 3.16



**Gambar 3.15** Flowchart Keseluruhan Sistem Alat



**Gambar 3.16** Flowchart Komunikasi Data Bluetooth

### 3.2.1 Perancangan Program Arduino

Perancangan program pada arduino ini berkaitan dengan pembuatan program untuk sub-sub sistem yang berupa sensor-sensor, aktuator, beberapa modul tambahan serta untuk pengiriman data oleh modul *bluetooth* dari alat untuk ditampilkan pada aplikasi *Android*. Langkah awal dalam suatu pemrograman adalah deklarasi variabel dan tipe data yang digunakan. Variabel dan tipe data ini digunakan untuk memudahkan dalam memprogram perangkat apa saja yang ingin diprogram. Pada Gambar 3.17 berikut merupakan deklarasi variabel dan tipe data yang akan digunakan :

```

int param=0;
int button_up,button_down,sel,metu=0;
char tampil[20];
int kolom=0;
int baris=0;
int temp;
int suhuS, suhuS1;
int lembabS, lembabS1;
int suhuP, suhuP1;
int lembabP, lembabP1;
int suhuB, suhuB1;
int lembabB, lembabB1;
int air, air1;
int state;
int lihat;
const int chipSelect = 4;
const int sensorw = A1;
int led=47;
float value1;
int humidity;
float volth;
float value2;

```

**Gambar 3.17** Deklarasi Variabel dan Tipe Data

Selanjutnya adalah pengaturan pada *void setup*, *void setup* digunakan untuk menginisialisasi pin digital yang kita pakai sebagai *input* atau *output*, inisialisasi komunikasi serial, *LCD Display*, serta beberapa perangkat modul pendukung seperti yang perlu di inisialisasikan yang terdapat pada Gambar 3.18 berikut :

---

```

void setup()
{
    pinMode(2, INPUT_PULLUP);
    pinMode(3, INPUT_PULLUP);
    pinMode(4, INPUT_PULLUP);
    pinMode(12, INPUT_PULLUP);
    pinMode(1ed, OUTPUT);
    pinMode(39, OUTPUT);
    pinMode(37, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(34, OUTPUT);
    pinMode(35, OUTPUT);
    lcd.begin(20, 4);
    lcd.createChar(0, subuu);
    lcd.createChar(1, anyeb);
    lcd.createChar(2, banyu);
    Serial.begin(9600);
    //Serial.flush();
    Wire.begin();
    RTC.begin();
    Serial.print("Initializing SD card...");
    if (!SD.begin(53)) {
        Serial.println("Card failed, or not present");
        return;
    }
    Serial.println("card initialized.");

    if (! RTC.isrunning()) {
        Serial.println("RTC is NOT running!");
        // following line sets the RTC to the date & time this sketch was compiled

```

---

**Gambar 3.18** Inisialisasi pada *Void Setup*

Pembuatan fungsi sensor HSM-20G digunakan untuk pembacaan temperatur dan kelembaban pada sensor HSM-20G. Dari program dibawah nilai kelembaban dan temperatur didapatkan dari hasil linierisasi sensor dari data yang telah diambil sebelumnya. Variabel *value1* adalah variabel yang digunakan untuk membaca nilai keluaran ADC pada pin A1. Variabel *volth* adalah konversi dari ADC ke Volt, dengan cara nilai hasil pembacaan ADC (*value1*) dikali Vref(4,3V) kemudian dibagi 1023. *Humidity* adalah hasil akhir dari linierisasi sensor kelembaban yang mempunyai formula nilai *volth* dikali dengan 30,8547580362547 kemudian dikurangi dengan 11,5038176856011 nilai tersebut didapatkan dari hasil perhitungan linierisasi data yang dijelaskan pada BAB IV. *Value2* adalah nilai pembacaan ADC pada pin analog A2. Variabel *temperature* adalah hasil akhir dari linierisasi sensor temperatur HSM-20G. Pada Gam-

bar 3.19 berikut merupakan program untuk linierisasi sensor HSM 20-G :

```
void hsm()
{
    value1 = analogRead(1); //Read data from analog pin and store it to value variable
    volth = ((value1*4.3)/1023);
    humidity = (((volth*30.8547580362547)-11.5038176856011));
    value2 = analogRead(2); //Read data from analog pin and store it to value variable
    temperature = (((value2*0.106068859)-37.90680477));
}
```

**Gambar 3.19** Fungsi Program HSM-20G

Selanjutnya pembuatan fungsi untuk pembacaan tanggal dan waktu pada modul RTC DS1307. Fungsi ini ditampilkan pada bagian menu awal pada LCD juga difungsikan untuk keperluan *datalogging*. Sebelum menggunakan RTC ini diatur terlebih dahulu tanggal dan waktu saat ini. Pada Gambar 3.20 merupakan program modul RTC DS1307.

```
void jam()
{
    lcd.clear();
    DateTime now = RTC.now();
    lcd.setCursor(0,0);
    lcd.print("Hidroponik Surabaya");
    lcd.setCursor(5, 1);
    lcd.print(now.day(), DEC);
    lcd.print('/');
    lcd.print(now.month(), DEC);
    lcd.print('/');
    lcd.print(now.year(), DEC);
    lcd.print(' ');
    lcd.setCursor(5, 2);
    if (now.hour()<10)
    lcd.print('0');
    lcd.print(now.hour(), DEC);
    lcd.print(':');
    if (now.minute()<10)
    lcd.print('0');
    lcd.print(now.minute(), DEC);
    lcd.print(':');
    if (now.second()<10)
    lcd.print('0');
    lcd.print(now.second(), DEC);
    lcd.setCursor(1,3);
    lcd.print(char(0));
    lcd.setCursor(2,3);
```

**Gambar 3.20** Fungsi Program Modul RTC DS1307



Pembuatan fungsi modul *SDCard* digunakan sebagai pembuatan *datalogger* yang digunakan untuk me-monitoring keadaan suatu *plant* yang sedang dikendalikan selain itu fungsi dari program ini juga digunakan untuk menyimpan set *point* parameter tiap tanaman yang selanjutnya akan dibandingkan dengan nilai yang sesuai dengan keadaan sebenarnya. Pada Gambar 3.21 merupakan program untuk SD Card.

---

```
void memori() {
    File dataFile = SD.open("3.txt", FILE_WRITE);
    dataFile.println("Set Point");
    if (kondisi==0)
    { dataFile.println("\tSuhu=" +String(suhuS1));}
    if (kondisi==1){
        dataFile.println("\tHumi="+String(lembabS1));
    }
    if (kondisi==2){
        dataFile.println("\tSuhu="+String(suhuP1));
    }
    if (kondisi==3){
        dataFile.println("\tHumi="+String(lembabP1));
    }
    if (kondisi==4){
        dataFile.println("\tSuhu="+String(suhuB1));
    }
    if (kondisi==5){
        dataFile.println("\tHumi="+String(lembabB1));
    }
    dataFile.println("SUHU&KELEMBABAN SEKARANG");
    dataFile.print("Suhu=");
    dataFile.println(temperature);
    dataFile.print("Lembab=");
    dataFile.println(humidity);
    delay(100);
    if (state==0){
        dataFile.println("SELADA");
    }
}
```

**Gambar 3.21** Fungsi Program SD Card

Dalam memasukkan nilai *set point* pada setiap parameter sayuran dibutuhkan suatu program yang berbeda dari sayuran lainnya berikut ini adalah contoh pembuatan fungsi program untuk memasukkan nilai set poin pada Temperatur Selada. Terdapat juga fungsi program untuk Temperatur Bayam dan Pakcoy serta Kelembaban Selada, Bayam dan Pakcoy semuanya memiliki fungsi

program yang hampir sama hanya berbeda nilai batasan maksimal yang dibutuhkan setiap sayuran dan nama dari fungsi yang digunakan. Berikut adalah penjelasan dari fungsi *SuhuSelada* yang digunakan untuk memasukkan nilai set poin suhu selada. Variabel *kondisi==0* adalah yang digunakan untuk mengindikasikan bahwa *kondisi==0* adalah Suhu Selada kondisi ini nantinya dipanggil kembali untuk dituliskan ke SD Card dan Bluetooth. Jika kondisi *button\_down==0* atau aktif maka *suhuS++* artinya adalah suhu yang bertipe data integer akan terus mencacah naik terus. *suhuS==31* digunakan untuk membatasi nilai atas dari masukan set poin ini adalah 31, dikarenakan tipe tombol yang digunakan *push release* maka ketika *button\_down==1* atau dalam keadaan tidak aktif akan menyimpan hasil akhir dari nilai yang dimasukkan pada *button\_down==0*. Variabel yang digunakan untuk menyimpan nilai akhir ini adalah *suhuS1*. Pada Gambar 3.22 merupakan program untuk memasukkan nilai *set point* untuk temperatur selada.

```
void SuhuSelada(){
    lcd.clear();
    kondisi==0;
    if (button_down==0){
        lcd.setCursor(0,0);
        lcd.print("Suhu=");
        lcd.setCursor(6,0);
        suhuS++;
        if (suhuS==31){
            suhuS=0;
            suhuS++;
        }
        lcd.print(suhuS);
        suhuS1=suhuS;
        delay(10);
    }
    if (button_down==1){
        lcd.setCursor(0,0);
        lcd.print("Suhu=");
        lcd.setCursor(6,0);
        lcd.print(suhuS1);
        delay(10);
    }
}
```

**Gambar 3.22** Fungsi Program Memasukkan Nilai *Set Point* untuk Temperatur Selada

Selanjutnya adalah pembuatan logika pemrograman menu pada alat ini, pembuatan logika ini menggunakan prinsip baris dan kolom seperti layaknya matriks. Setiap baris dan kolom merepresentasikan letak dan posisi pilihan dari menu, untuk perpindahan pilihan menu digunakan variabel *temp* untuk membantu dalam pilihan menu atau lebih mudahnya *temp* sama dengan baris. Kemudian untuk sinkronisasi pilihan menu dan *push button*. Pada pilihan *button down* bekerja jika  $temp > 0$  maka *temp* akan berkurang 1. Pada pilihan *button up* bekerja jika  $temp < 3$  maka *temp* akan bertambah 1. Pada pilihan *button down* bekerja jika  $temp > 0$  maka *temp* akan berkurang 1. Pada pilihan *Selector* bekerja jika  $kolom < 7$  maka  $kolom = kolom + 1$  dan  $baris = temp$ . Pada pilihan *Back* bekerja jika  $kolom > 1$  maka  $kolom = kolom - 1$  dan  $baris = temp = 0$ . Pada Gambar 3.23 merupakan program menu dan pada Gambar 3.24 merupakan program untuk mengaktifkan aktuator.

---

```

void loop()
{
    bt();
    if(kolom==0&&baris==0){
        lcd.setCursor(5,0);
        lcd.print("Hidroponik");
        delay(100);
    }
    if(kolom==1&&baris==0){
        jam();
    }
    if(kolom==2&&baris==0){
        if(temp==0){
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(">Pilih Sayur");
            lcd.setCursor(0,1);
            lcd.print(" Aktifkan Humidifier");
            lcd.setCursor(0,2);
            lcd.print(" Aktifkan Pompa");
            delay(100);
        }
        else if(temp==1){
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(" Pilih Sayur");
            lcd.setCursor(0,1);
            lcd.print(">Aktifkan Humidifier");
            lcd.setCursor(0,2);
            lcd.print(" Aktifkan Pompa");
        }
    }
}

```

---

**Gambar 3.23** Program Menu Sesuai Baris dan Kolom

```

if(kolom==3&&baris==1){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("HUMI ON");
    delay(100);
    digitalWrite(49, LOW);
    delay(100);
}
else {digitalWrite(49,HIGH);}
if(kolom==3&&baris==2){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("pompa ON");
    digitalWrite(48, HIGH);
    delay(100);
}
}

```

**Gambar 3.24** Program Pengaktifan Aktuator

```

void bt() {
    if (state==0) {
        Serial3.print("SELADA");
        Serial3.print("|");
        Serial3.print(suhuS1);
        Serial3.print("|");
        Serial3.print(temperature);
        Serial3.print("|");
        Serial3.print(lembabS1);
        Serial3.print("|");
        Serial3.print(humidity);
    }
}

```

**Gambar 3.25** Program Komunikasi Data *Bluetooth* dengan *Android*

Pada Gambar 3.25 merupakan program pada *software Arduino* untuk komunikasi *bluetooth* dengan *Android*. Pada program tersebut terletak pada sebuah *void* yang berisi 3 status yaitu status 0, 1, dan 2. Jika *state/status* sama dengan 0 maka akan menampilkan parameter untuk sayuran selada. Sedangkan untuk status sama dengan 1 maka menampilkan parameter sayuran pakcoy, dan status sama dengan 2 untuk menampilkan parameter sayuran bayam. Dengan menggunakan fungsi *Serial3.print* yang berarti menampilkan parameter yang terdapat pada alat. *Serial3.print("SELADA")* untuk menampilkan nama sayuran selada, *Serial3.print(suhuS1)* untuk menampilkan *set point* temperatur dari selada, *Serial3.print(temperature)* untuk menampilkan temperatur yang terbaca oleh sensor, *Seri-*

*al3.print(lembabS1)* untuk menampilkan *set point* kelembaban dari selada, dan *Serial3.print(humidity)* untuk menampilkan kelembaban yang terbaca oleh sensor. Setiap tampilan dari parameter tersebut dipisahkan dengan tanda “ | ” dengan program *Serial3.print("|")* yang berfungsi untuk memisahkan data agar tidak terjadi penumpukan data.

```
if (Serial3.available()>0) {
    serialA =Serial3.read();
    Serial3.println(serialA);
}
switch(serialA){
    case 49:
        digitalWrite(pinPompa, HIGH);
        break;

    case 50:
        digitalWrite(pinPompa, LOW);
        break;
```

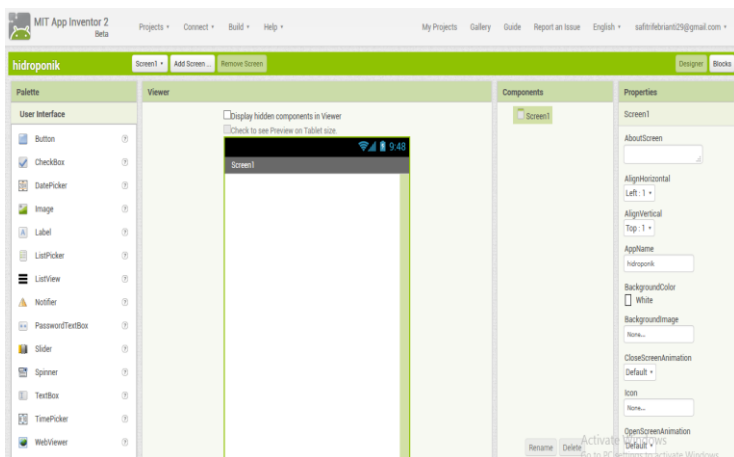
**Gambar 3.26** Program *Button* Aplikasi *Android*

Pada Gambar 3.26 merupakan program *button* arduino untuk aplikasi *android*. Program tersebut menggunakan komunikasi serial nomor 3 pada arduino. *Serial3.available* merupakan inisialisasi bahwa komunikasi serial nomor 3. Pada program ini menggunakan perintah *switch case* dan menggunakan *serialA* yang merupakan inisialisasi dari kode data bit yang akan digunakan. Apabila kedua perangkat *bluetooth* telah terpasang dan melakukan eksekusi terhadap aplikasi dengan kode data yang dikirimkan sama dengan 49, maka pin untuk keluaran pompa akan bernilai *HIGH* yang berarti pompa menyala. Sebaliknya, jika kode yang dikirimkan sama dengan 50, keluaran pompa bernilai *LOW*, yang berarti pompa dalam keadaan mati. Pada program ini juga menggunakan keluaran untuk *humidifier* dengan menggunakan kode data 51 dan 52.

### 3.2.2 Perancangan Program Aplikasi *Android*

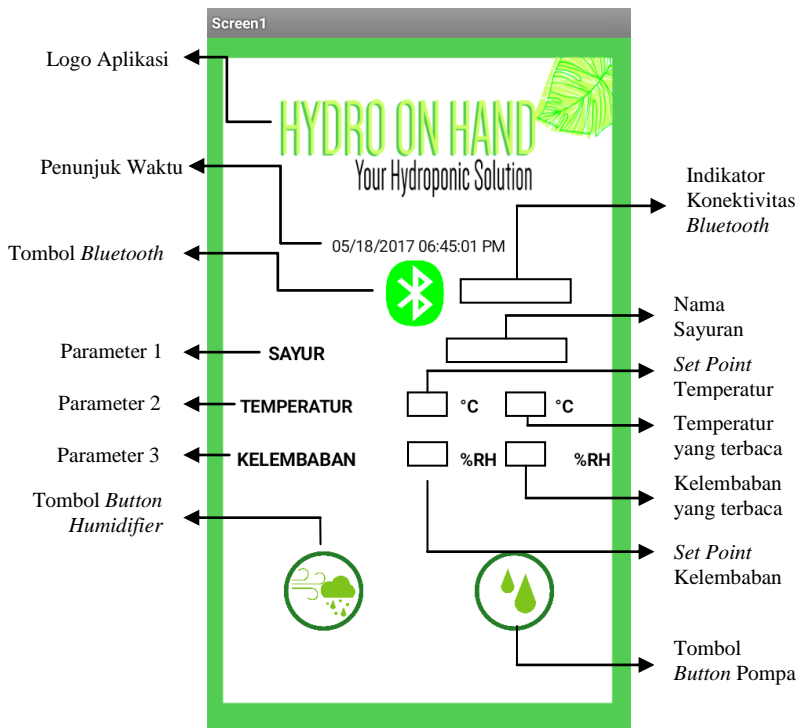
Aplikasi *Android* digunakan untuk *user interface* pada pengguna ataupun pemilik kebun hidroponik. Aplikasi ini bertujuan untuk memonitoring kondisi dari parameter-parameter tanaman yang sedang diuji. *Software* yang digunakan untuk perancangan aplikasi *Android* menggunakan MIT App Inventor dimana pembuatan pro-

gram dilakukan secara *online* pada alamat *web* <http://appinventor.mit.edu/> dan harus *login* menggunakan akun *Gmail*. Selanjutnya akan tampil *Designer View* kosong seperti pada Gambar 3.27 yang nantinya digunakan untuk menyusun komponen dan mendesain *layout user interface* pada sebuah aplikasi. Komponen-komponen tersebut berada pada sisi kiri dan untuk menambahkannya pengguna cukup melakukan *drag and drop* komponen ke *project*. Penggunaan fungsi dari komponen tersebut tergantung dengan kebutuhan.



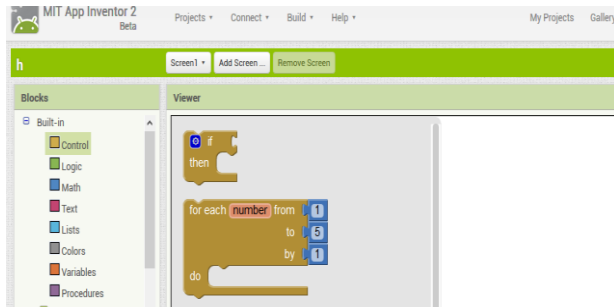
**Gambar 3.27** Tampilan *Designer View*

Pada aplikasi ini terdapat satu tampilan utama yang berisi konektivitas antara *bluetooth* pada *smartphone Android* dengan modul *bluetooth* HC-05 pada alat. Selain itu, juga terdapat tampilan monitoring jenis sayur yang ditanam, temperatur dan kelembaban, serta terdapat dua buah *button* untuk menyalakan atau mematikan *humidifier* dan atau pompa air. Tampilan dari aplikasi *Android* ditunjukkan pada Gambar 3.28 :



**Gambar 3.28** Tampilan *Layout* Aplikasi Android

Setelah membuat *layout* aplikasi yang diinginkan, barulah memasukkan program berupa blok-blok kode dengan cara *drag and drop* blok-blok kode tersebut pada *blocks view*. Tampilan dari *block view* terdapat pada Gambar 3.29 berikut :



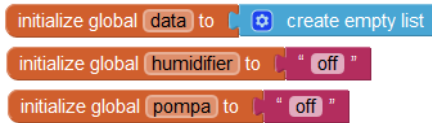
**Gambar 3.29** Tampilan *Blocks View*

Selanjutnya, menyusun blok-blok yang terdapat pada bagian *Built-in*. Tersedia 7 pilihan blok yang nantinya akan kita pilih sesuai dengan kebutuhan aplikasi yang dibuat. Rincian penyusunan program dari blok tersebut terdapat pada Gambar 3.30 sampai dengan 3.38. Pada Gambar 3.30 merupakan program inisialisasi variabel global yang diberi nama “data”. Isi dari variabel global ini diawali dengan menuliskan *list* kosong yang nantinya akan diisi oleh banyaknya *list* data yang dikirim oleh Arduino ke *smartphone Android*. Selain itu terdapat juga blok program inisialisasi yang menyatakan aktuator-aktuator yang digunakan diantaranya *humidifier* dan pompa air.

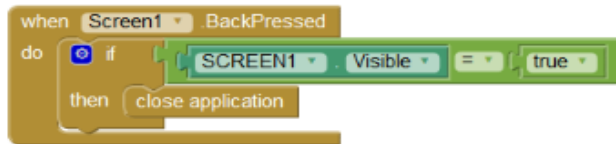
Pada Gambar 3.31 merupakan program untuk menutup aplikasi. Dengan menggunakan blok *Back Pressed*, apabila *Screen 1* berlogika benar saat *visible* (terlihat) maka pada saat tombol kembali pada *smartphone Android* ditekan, aplikasi akan tertutup secara otomatis. Pada Gambar 3.32 merupakan program *List Picker Bluetooth*, yang apabila tombol *List Picker Bluetooth* ditekan maka akan menampilkan alamat dari perangkat *bluetooth* yang akan dipasangkan dengan *smartphone Android*. Pada Gambar 3.33 adalah program konektivitas *bluetooth*. Apabila telah menekan tombol *bluetooth* dan telah memilih perangkat *bluetooth*, maka program akan memanggil *Bluetooth Client* untuk menghubungkan dengan perangkat *bluetooth* yang telah dipilih. Apabila perangkat *bluetooth* pada *smartphone* telah terhubung dengan perangkat *bluetooth* pada alat maka label akan menampilkan keterangan “*Connected*” dengan tulisan berwarna biru. Namun, apabila perangkat *bluetooth* pada *smartphone* tersebut tidak terhubung dengan perangkat *bluetooth*



pada alat, maka *label* akan menampilkan keterangan “*Not Connected*” dengan tulisan berwarna merah.



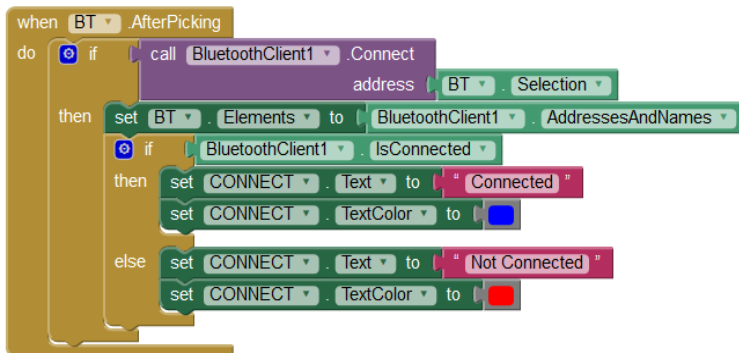
**Gambar 3.30** Blok Program Inisialisasi Variabel Data



**Gambar 3.31** Blok Program *Close Application*

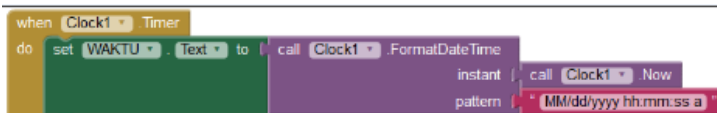


**Gambar 3.32** Blok Program *List Picker Bluetooth*

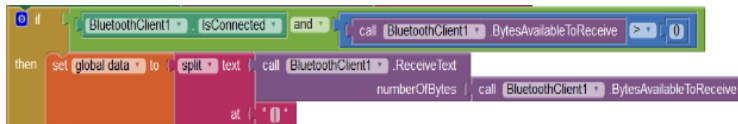


**Gambar 3.33** Blok Program Konektivitas *Bluetooth*

Selanjutnya, terdapat blok program yang berisi *Clock1* yang merupakan komponen utama pada program ini. Pada Gambar 3.34 merupakan blok program untuk menampilkan waktu. Blok tersebut menggunakan *timer* dengan nama “*Clock1*” yang apabila timer telah di eksekusi maka akan menampilkan waktu dengan format “MM/dd/yyyy hh:mm:ss” (“bulan/tanggal/tahun jam:menit:detik”) dan menampilkannya dalam keadaan *real time* pada saat aplikasi dibuka. Pada Gambar 3.35 merupakan blok penerimaan data dimana blok tersebut masih berada satu blok sebelumnya. Apabila *bluetooth* sudah terhubung, dan data yang diterima dari Arduino lebih dari 0, maka jadikan isi list dari variabel global data menjadi hasil data dari Arduino yang telah di pisahkan dengan tanda “|”. Data-data tersebut nantinya akan ditampilkan pada tampilan aplikasi secara *real time*.

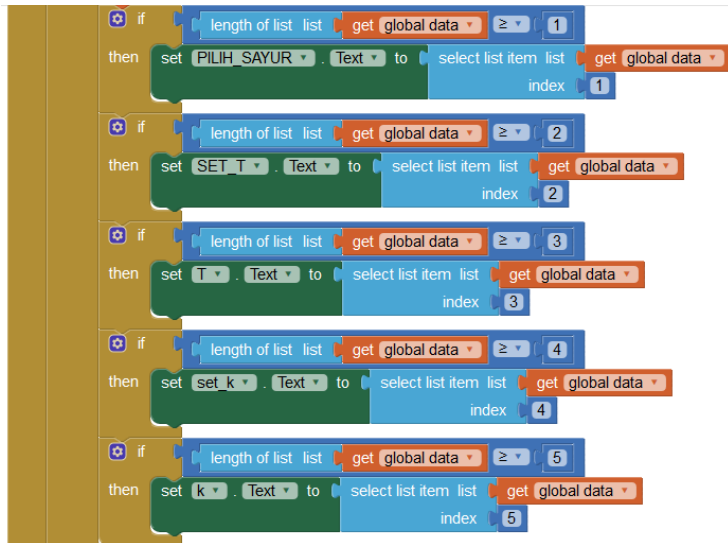


**Gambar 3.34** Blok Program Tampilan Waktu



**Gambar 3.35** Blok Program Penerimaan Data

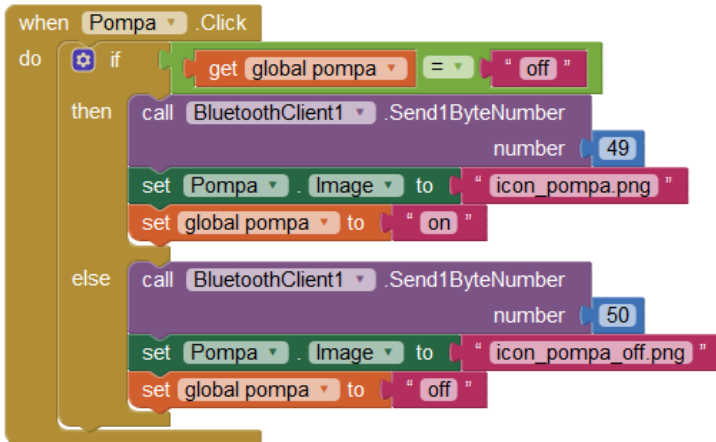
Masih berada dengan satu blok dengan blok sebelumnya, Gambar 3.36 merupakan blok program untuk menampilkan nama sayur yang ditanam pada alat. Hasil dari tampilan tersebut sesuai dengan nilai *item list* pada variabel data dengan *index list* 1. Terdapat tiga pilihan sayur yang dapat dipilih pada alat yang selanjutnya nama dari sayur yang dipilih akan ditampilkan pada aplikasi *smartphone*. Pada *index list* 2, menampilkan *set point* temperatur dari sayur yang dipilih dan sesuai dengan *set point* temperatur yang telah diatur pada alat. Pada *index list* 3, menampilkan nilai temperatur dari *plant* yang terbaca oleh sensor HSM-20G. Pada *index list* 4, menampilkan *set point* kelembaban dari sayur yang dipilih dan sesuai dengan *set point* kelembaban yang telah diatur pada alat. Dan pada *index list* 5, menampilkan nilai kelembaban dari *plant* yang terbaca oleh sensor HSM-20G.



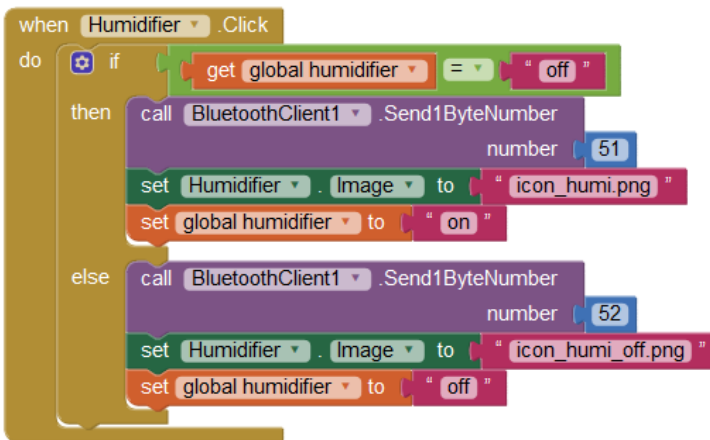
**Gambar 3.36** Blok Program Menampilkan Nama Sayur

Pada Gambar 3.37 merupakan blok program *button* untuk mengaktifkan atau menonaktifkan pompa. Apabila *bluetooth* dalam keadaan terhubung dan tombol tersebut ditekan maka akan mengirimkan kode ASCII 49 yang berarti 1, sehingga pompa pada alat akan aktif dan secara otomatis ikon *button* berwarna hijau. Sedangkan untuk menonaktifkan pompa, dengan menekan tombol tersebut maka akan mengirimkan kode ASCII 50 yang berarti 2, dan secara otomatis ikon *button* berubah warna menjadi abu-abu yang menandakan pompa tidak aktif.

Pada Gambar 3.38 merupakan blok program *button* untuk mengaktifkan atau menonaktifkan *humidifier*. Apabila *bluetooth* dalam keadaan terhubung dan tombol tersebut ditekan maka akan mengirimkan kode ASCII 51 yang berarti 3, sehingga *humidifier* pada alat akan aktif dan secara otomatis ikon *button* berwarna hijau. Sedangkan untuk menonaktifkan *humidifier*, dengan menekan tombol tersebut maka akan mengirimkan kode ASCII 52 yang berarti 4, dan secara otomatis ikon *button* berubah warna menjadi abu-abu yang menandakan *humidifier* tidak aktif.



**Gambar 3.37** Blok Program *Button Pompa*



**Gambar 3.38** Blok Program *Button Humidifier*

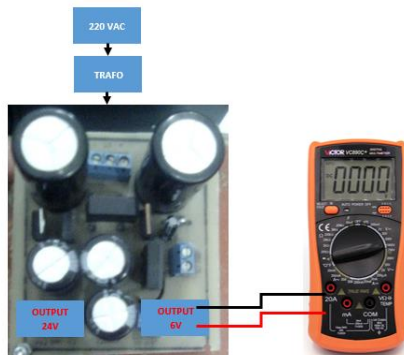
## BAB IV

### PENGUJIAN DAN ANALISA DATA

Untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini telah terlaksana atau tidak, perlu dilakukan pengujian dan analisa terhadap alat yang dibuat. Pengujian dibagi menjadi dua bagian sesuai dengan tugas masing-masing mahasiswa. Moh. Ilham Aziz W. melakukan pengujian pada poin 4.1 sampai dengan 4.6, sedangkan Safitri Febrianti melakukan pengujian 4.7 sampai dengan 4.12. setelah dilakukan pengujian, dilakukan analisa terhadap bagian-bagian alat yang telah diuji.

#### 4.1 Pengujian Rangkaian *Power Supply*

Pengujian *power supply* 6 dan 24 V DC bertujuan untuk mengetahui tegangan keluaran yang dapat digunakan untuk tugas akhir ini. Cara pengujian *power supply* dilakukan dengan mengukur tegangan *output* 6 V dan 24 V menggunakan AVometer digital seperti pada Gambar 4.1 berikut :



**Gambar 4.1** Metode Pengujian *Power Supply*

Selanjutnya, untuk mengetahui apakah *power supply* dapat digunakan atau tidak, dengan cara menghitung presentase *error* dengan rumus pada Persamaan 4.1 dan menghasilkan data pada Tabel 4.1 :

$$\% \text{ Error} = \left| \frac{(\text{NilaiAcuan} - \text{NilaiPengukuran})}{\text{NilaiAcuan}} \right| \times 100\% \dots\dots\dots (4.1)$$

**Tabel 4.1** Data Pengujian *Power Supply*

<i>Power Supply</i>	<i>Output (V)</i>	<i>Error(%)</i>
+6 V	6,11	1,83
+24 V	23,89	0,45

Pengujian dilakukan dengan mengukur keluaran dengan volt-meter, kemudian dihitung presentasi *error*.

$$\% \text{ Error} = \left| \frac{(\text{NilaiAcuan} - \text{NilaiPengukuran})}{\text{NilaiAcuan}} \right| \times 100\%$$

$$\% \text{ Error} = \left| \frac{(6 - 6,11)}{6} \right| \times 100\% = 1,83\%$$

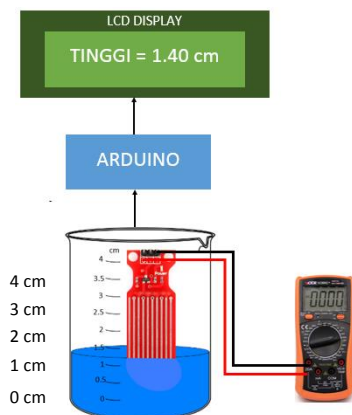
$$\% \text{ Error} = \left| \frac{(24 - 23,89)}{24} \right| \times 100\% = 0,45\%$$

Dari hasil tegangan diatas dapat diketahui bahwa pada *supply* +6 V nilai ini adalah nilai acuan *voltage regulator* pada IC 7806, output tegangan yang terukur sampai ke titik kestabilan mencapai +6,11 V nilai ini didapatkan dengan mengukur keluaran *supply* menggunakan AVO meter. Pada *supply* +24 V nilai ini adalah nilai acuan *voltage regulator* pada IC 7824, *output* tegangan berhenti di 23,89 V, nilai ini didapatkan dengan mengukurnya dengan AVO meter. Maka dari itu, *power supply* ini dapat digunakan dalam sistem ini untuk men-*supply* komponen atau instrumen dalam sistem ini.

## 4.2 Pengujian Sensor *Water Level* Funduino

Pengujian sensor *water level* ini dilakukan dengan cara menggunakan sebuah gelas ukur yang pada sisi dalam nya lah ditempelkan sensor ketinggian air. Kemudian, dimasukkan air secara bertahap mulai dari 0,5 cm sampai dengan ketinggian maksimal dari sensor yaitu 4 cm. Karena keluaran dari sensor ini merupakan tegangan, maka dilakukan pengukuran tegangan keluaran (*Vout*) menggunakan AVOMeter digital pada kaki data sensor (S) dan kaki *ground* (-) untuk setiap ketinggian yang dicapai sensor. Pengujian ini

dilakukan seperti pada Gambar 4.2 dan menghasilkan data seperti pada Tabel 4.2 berikut :



**Gambar 4.2** Metode Pengujian Sensor Ketinggian Air

**Tabel 4.2** Data Pengujian Sensor Ketinggian Air

No.	Ketinggian Air (cm)	Vout (V)
1.	0,5	2,06
2.	1	2,19
3.	1,5	2,29
4.	2	2,34
5.	2,5	2,39
6.	3	2,43
7.	3,5	2,52

Setelah di dapatkan data tersebut, dengan menggunakan metode regresi linier *slope intercept*, maka diperoleh fungsi seperti pada Persamaan 4.2 berikut :

$$y = 1,218905053x - 0,565117967 \dots\dots\dots(4.2)$$

Keterangan :

y = ketinggian air (cm)

x = Vout (V)

Selanjutnya, untuk mengetahui sensor tersebut telah linier, dilakukan pengambilan data ulang dengan menggunakan program Arduino seperti pada Gambar 4.3.

```

void water () {
  val = analogRead (sensorw);
  float volts = ((val*4.4)/1023);
  h=(volts*1.218905053)-0.565117967;
  delay (1000);
}

```

**Gambar 4.3** Program Pengujian Sensor Ketinggian Air

Pengambilan data dilakukan dengan menggunakan *power supply* 5 V dengan *display* LCD serta dilakukan perbandingan antara ketinggian yang mengenai sensor dengan ketinggian yang dicapai gelas ukur seperti pada Gambar 4.1. Selanjutnya dihitung pula nilai % *error* dengan rumus pada Persamaan 4.1 sebagai contoh :

$$\% \text{ Error} = \left| \frac{(0,5 - 2,05)}{0,5} \right| \times 100\% = 310\%$$

Dari hasil perhitungan diatas, dapat dianalisa bahwa pada ketinggian air yang sebenarnya adalah 0,5 cm, namun ketinggian air yang dibaca oleh sensor adalah 2,05 cm sehingga didapatkan persentase *error* sebesar 310 %. Selanjutnya, dengan cara dan rumus yang sama, dihitung persentase *error* untuk ketinggian 1 cm sampai dengan 4 cm. Pada Tabel 4.3 berikut merupakan data perbandingan ketinggian air yang terbaca oleh sensor dengan ketinggian air sebenarnya.

**Tabel 4.3** Data Perbandingan Ketinggian Air

No.	Ketinggian Air Sebenarnya (cm)	Ketinggian Air yang terbaca oleh sensor (cm)	% <i>Error</i>
1.	0,5	2,05	310%
2.	1	2,2	120%
3.	1,5	2,3	53,3%
4.	2	2,4	20%
5.	2,5	2,49	0,4%
6.	3	2,96	1,3%
7.	3,5	3,01	14%
8.	4	3,16	21%

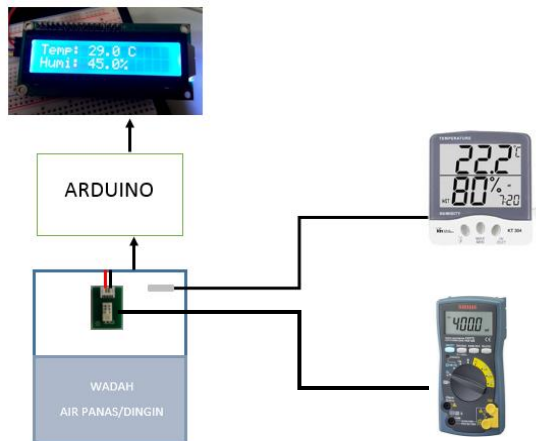
Pada Tabel 4.3, pada ketinggian 0,5 sampai dengan 2 cm, % *error* sangat tinggi. Namun pada ketinggian 2,5 sampai dengan 3 cm % *error* terlampaui kecil yaitu hanya sekitar 0,4 – 1,3 %. Pada alat ini



yang digunakan adalah ketinggian 2,5 sampai dengan 3 cm karena menyesuaikan dengan kebutuhan tanaman.

### 4.3 Pengujian Sensor HSM 20-G

Pada pengujian sensor HSM-20G terdapat dua sub bab, yaitu pengujian temperatur dan pengujian kelembaban. Pengujian sensor ini dilakukan dengan cara menempatkan sensor pada wadah tertutup berisi air sehingga udara pada wadah tersebut dapat di deteksi oleh sensor. Selain itu, untuk mengetahui nilai temperatur dan kelembaban, dibutuhkan alat pengukur temperatur dan kelembaban (*thermo-hygrometer*) bertipe HTC-2. Karena keluaran dari sensor ini merupakan tegangan, maka dilakukan pengukuran tegangan keluaran (Vout) menggunakan AVometer digital pada kaki data temperatur (T) dengan kaki *ground* (-) serta pada kaki data kelembaban (H) dengan kaki *ground* (-). Pada Gambar 4.4 berikut merupakan metode untuk melakukan pengujian sensor HSM 20-G:



**Gambar 4.4** Metode Pengujian Sensor HSM 20-G

#### 4.3.1 Pengujian Temperatur

Dengan menggunakan metode seperti pada Gambar 4.3, diperoleh data pengujian temperatur pada Tabel 4.4 sebagai berikut :

**Tabel 4.4** Data Pengujian Temperatur Sensor HSM 20-G

No.	Temperatur (°C)	Vout (V)
1.	20	2,05

No.	Temperatur (°C)	Vout (V)
2.	21,1	2,085
3.	22	2,097
4.	23,1	2,113
5.	24,4	2,142
6.	25,4	2,163
7.	26,5	2,233
8.	27	2,27
9.	28	2,32
10.	29	2,36
11.	30,1	2,4
12.	31	2,45
13.	32,1	2,465
14.	33	2,53
15.	34,4	2,54
16.	35,2	2,567

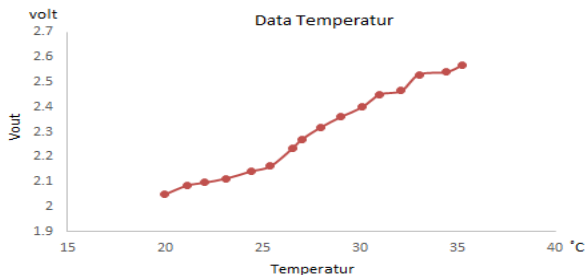
Setelah di dapatkan data, dengan menggunakan metode regresi linier *slope intercept*, maka diperoleh Persamaan 4.3 dan diperoleh grafik pada Gambar 4.5 sebagai berikut :

$$y = 26,42793994x - 33,11573568 \dots\dots\dots(4.3)$$

Keterangan :

y = temperatur (°C)

x = Vout (V)



**Gambar 4.5** Data Temperatur

Selanjutnya, untuk mengetahui sensor dianggap linier, maka dilakukan pengambilan data ulang dengan menggunakan program Arduino yang mengacu pada Gambar 3.19. Pengambilan data di

lakukan dengan menggunakan *power supply* 5 V dengan *display* LCD serta dilakukan perbandingan antara temperatur yang terbaca oleh sensor dengan alat ukur *thermohygrometer* seperti pada Gambar 4.4. Selanjutnya dihitung pula nilai % *error* dengan rumus pada Persamaan 4.1 sebagai contoh :

$$\% \text{ Error} = \left| \frac{(22,9 - 23,4)}{22,9} \right| \times 100\% = 2,18\%$$

Dari hasil perhitungan, dapat dianalisa bahwa pada temperatur pada alat ukur adalah 22,9 °C, namun temperatur yang terbaca oleh sensor adalah 23,4 °C sehingga didapatkan persentase *error* sebesar 2,18 %. Selanjutnya, dengan cara dan rumus yang sama, dihitung persentase *error* untuk temperatur sampai dengan 35,3 °C. Pada Tabel 4.5 berikut merupakan data perbandingan temperatur yang terbaca oleh sensor dengan temperatur pada alat ukur HTC-2.

**Tabel 4.5** Data Perbandingan Temperatur

No.	Temperatur pada alat ukur HTC-2 (°C)	Temperatur yang terbaca oleh sensor (°C)	% Error
1.	22,9	23,4	2,18%
2.	23,5	24,4	3,82%
3.	25,5	25,7	0,78%
4.	26,0	26,7	2,69%
5.	26,9	27,1	0,74%
6.	27,5	27,9	1,45%
7.	28,2	28,1	0,34%
8.	28,4	28,6	0,7%
9.	29,2	29,6	1,37%
10.	29,8	29,9	0,33%
11.	30,6	30,6	0,65%
12.	31,8	31,6	0,62%
13.	32,7	32,2	1,52%
14.	35,3	33,5	3,39%

4.3.2 Pengujian Kelembaban

Dengan menggunakan metode seperti pada Gambar 4.4, diperoleh data pengujian kelembaban seperti pada Tabel 4.6 sebagai berikut :

Tabel 4.6 Data Pengujian Kelembaban Sensor HSM 20-G

No.	Kelembaban (%RH)	Vout (V)
1.	10	0,74
2.	20	0,95
3.	30	1,31
4.	40	1,68
5.	50	2,02
6.	60	2,37
7.	70	2,69
8.	80	2,99
9.	90	3,19

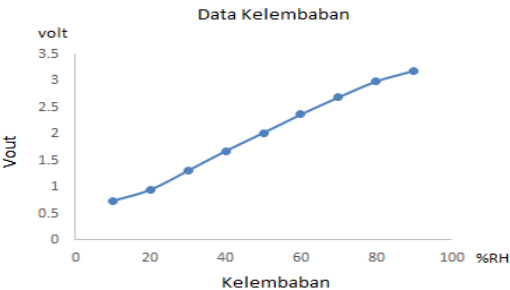
Setelah di dapatkan data tersebut, dengan menggunakan metode regresi linier *slope intercept*, maka diperoleh Persamaan 4.4 dan diperoleh grafik pada Gambar 4.6 sebagai berikut :

$$y = 30,85475804x-11,50381769 \dots\dots\dots(4.4)$$

Keterangan :

x = kelembaban (%RH)

y = Vout (V)



Gambar 4.6 Data Kelembaban

Selanjutnya, untuk mengetahui sensor tersebut telah dianggap linier, dilakukan pengambilan data ulang dengan menggunakan program Arduino yang mengacu pada Gambar 3.19. Pengambilan data dilakukan dengan menggunakan *power supply* 5 V dengan *display* LCD serta dilakukan perbandingan antara kelembaban yang terbaca oleh sensor dengan alat ukur *thermohygrometer* seperti pada Gambar 4.4. Selanjutnya dihitung pula nilai % *error* dengan rumus pada Persamaan 4.1 sebagai contoh :

$$\% \text{ Error} = \left| \frac{(50 - 49)}{50} \right| \times 100\% = 2\%$$

Dari hasil perhitungan, dapat dianalisa bahwa pada kelembaban pada alat ukur adalah 50 %RH, namun kelembaban yang terbaca oleh sensor adalah 49 %RH, sehingga didapatkan persentase *error* sebesar 2 %. Selanjutnya, dengan cara dan rumus yang sama, dihitung persentase *error* untuk kelembaban sampai dengan 88 %RH. Pada Tabel 4.7 berikut merupakan data perbandingan kelembaban yang terbaca oleh sensor dengan kelembaban pada alat ukur HTC-2.

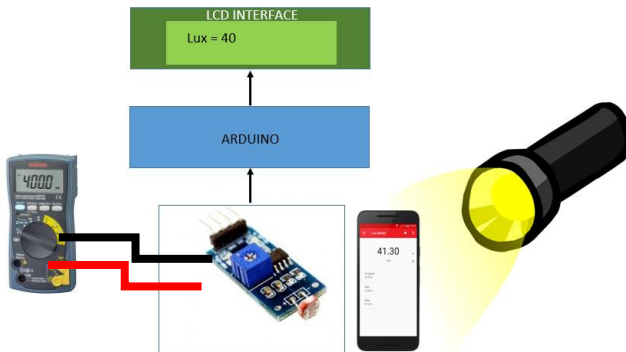
**Tabel 4.7** Data Perbandingan Kelembaban

No.	Kelembaban pada alat ukur HTC-2 (%RH)	Kelembaban yang terbaca oleh sensor (%RH)	% Error
1.	50	49	2%
2.	55	53	3,63%
3.	57	56	1,75%
4.	60	59	2,69%
5.	66	65	1,51%
6.	68	66	1,45%
7.	71	70	1,40%
8.	74	73	1,35%
9.	76	75	1,37%
10.	79	79	0%
11.	80	79	1,25%
12.	84	82	2,38%
13.	86	85	1,16%
14.	88	87	1,1%

Pada Tabel 4.5 dan Tabel 4.7 dapat dilihat bahwa persentase *error* terlampau kecil. Dengan demikian, sensor HSM 20-G dapat digunakan untuk sensor temperatur dan kelembaban pada tugas akhir ini.

#### 4.4 Pengujian Sensor Cahaya

Pengujian sensor cahaya LDR dilakukan dengan memberikan cahaya terang sampai dengan gelap. Untuk cahaya yang terang dengan cara mendekatkan dan menjauhkan lampu senter pada sensor, sedangkan untuk kondisi redup hingga gelap dengan cara menutup sensor dengan sebuah benda. Dari perlakuan tersebut diperoleh nilai keluaran ADC pada mikrokontroler Arduino dan dibandingkan nilainya dengan nilai fluks pada aplikasi *Flux Meter* pada *smartphone Android*. Berikut ilustrasi metode pengujian pada Gambar 4.7 dan data hasil pengujian pada Tabel 4.8:



**Gambar 4.7** Metode Pengujian Sensor Cahaya

**Tabel 4.8** Pengukuran Sensor Cahaya

LUX	ADC
3	960
9	908
21	685
33	668
79	601
93	612

LUX	ADC
113	602
123	573
159	499
169	329
363	341
769	171
1415	147
1821	121
2500	105
3708	72

Setelah di dapatkan data tersebut, dengan menggunakan metode regresi linier *slope intercept*, maka diperoleh Persamaan 4.5 sebagai berikut:

$$y = 1937,47054x - 2,81769942 \dots\dots\dots(4.5)$$

Keterangan :

y = Fluks (Cd)

x = ADC

Selanjutnya, untuk mengetahui sensor tersebut telah dianggap linier, dilakukan pengambilan data ulang dengan menggunakan program Arduino seperti pada Gambar 4.8 :

```

val = analogRead (analogPin);
lux = ((val*1937,47054)-2,81769942);
Serial.println(val);
lcd.setCursor(0,0);
lcd.print(lux);
delay (1000);
}

```

**Gambar 4.8** Program Pengujian Sensor Cahaya

Pengambilan data dilakukan dengan menggunakan *power supply* 5 V dengan *display* LCD serta dilakukan perbandingan antara kelembaban yang terbaca oleh sensor dengan alat ukur berupa aplikasi *Flux Meter* seperti pada Gambar 4.7. Selanjutnya dihitung pula nilai % *error* dengan rumus pada Persamaan 4.1 sebagai contoh :

$$\% \text{ Error} = \left| \frac{(5-8)}{5} \right| \times 100\% = 60\%$$

Dari hasil perhitungan, dapat dianalisa bahwa nilai fluks pada alat ukur adalah 5 Cd, namun kelembaban yang terbaca oleh sensor adalah 8 Cd, sehingga didapatkan persentase *error* sebesar 60 %. Selanjutnya, dengan cara dan rumus yang sama, dihitung persentase *error* untuk fluks sampai dengan 2795. Pada Tabel 4.9 berikut merupakan data perbandingan fluks yang terbaca oleh sensor dengan nilai fluks pada alat ukur *flux meter*.

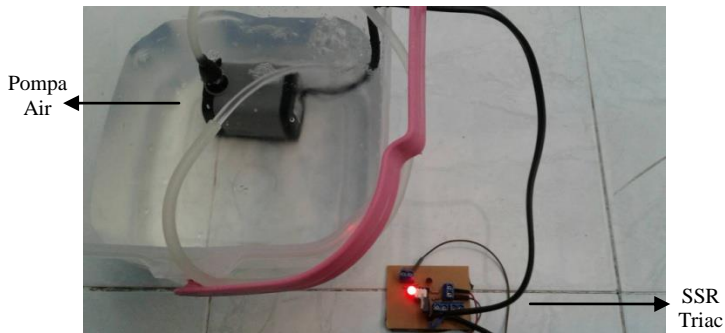
**Tabel 4.9** Data Perbandingan Cahaya

No.	Fluks pada Aplikasi <i>Flux Meter</i> (Cd)	Fluks yang terbaca oleh sensor (Cd)	% <i>Error</i>
1.	5	8	60 %
2.	19	17	10,5 %
3.	80	90	11,1 %
4.	165	198	16,7 %
5.	349	378	7,6%
6.	563	590	5,86%
7.	893	925	3,58%
8.	1298	1354	4,31%
9.	2795	3098	10,8%

#### 4.5 Pengujian Pompa Air dengan *Switching TRIAC*

Pengujian ini dilakukan agar dapat mengetahui *Solid State Relay* (TRIAC) dapat bekerja sesuai sistem agar pompa air dapat terkontrol dengan baik. Metode pengujian rangkaian *Solid State Relay* (TRIAC) dilakukan dengan menerapkan program sederhana pada mikrokontroler Arduino yang mengacu pada Gambar 3.24. Status ON pada *Solid State Relay* (TRIAC) diperoleh jika basis pada transistor BC337 diberi *input logic* 1 dari mikrokontroler atau diberi tegangan +5 V yang telah memberikan logika aktif pada *optocoupler* PC817 dan pompa akan aktif. Sedangkan apabila basis pada transistor BC337 diberi *input* 0 dari mikrokontroler arduino atau diberi tegangan 0 V, maka *Solid State Relay* (TRIAC) akan berganti status menjadi OFF dan pompa akan mati. Pengujian TRIAC dan pompa air dapat dilihat pada Gambar 4.9 berikut :

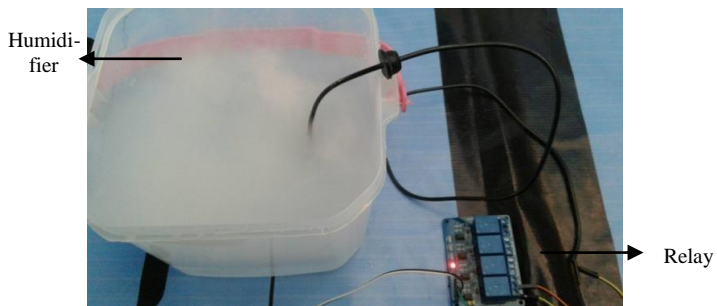




**Gambar 4.9** Pengujian Pompa Air dengan *Switching* SSR TRIAC

#### **4.6 Pengujian *Ultrasonic Humidifier* dengan *Switching Relay***

Pengujian ini dilakukan agar dapat mengetahui *relay* dapat bekerja sesuai sistem agar *ultrasonic humidifier* dapat terkontrol dengan baik. Metode pengujian rangkaian *relay* dilakukan dengan menerapkan program sederhana pada mikrokontroler Arduino yang mengacu pada Gambar 3.24. Status ON pada *relay* diperoleh jika basis pada transistor BD139 diberi *input logic* 0 dari mikrokontroler atau diberi tegangan 0 V yang telah memberikan *logic* nonaktif pada *optocoupler* PC817 dan menyebabkan *ultrasonic humidifier* nyala. Sedangkan apabila basis pada transistor BD139 diberi *input* 1 dari mikrokontroler Arduino atau diberi tegangan +5 V, maka *relay* akan berganti status menjadi ON dan menyebabkan *ultrasonic humidifier* akan mati. Pengujian dapat dilihat pada Gambar 4.10 berikut :



**Gambar 4.10** Pengujian *Ultrasonic Humidifier* dengan *Relay*

#### 4.7 Pengujian LCD (*Liquid Crystal Display*) 20x4

Pengujian LCD ini dilakukan untuk mengetahui tampilan dari sistem pada alat ini bekerja sesuai sistem. Pengujian dilakukan dengan mikrokontroler Arduino dengan menuliskan karakter “Hidroponik” pada LCD seperti pada Gambar 4.10 sehingga didapatkan tampilan seperti pada Gambar 4.11 berikut :

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  lcd.begin(20, 4);
  lcd.print("Hidroponik");
}
```

**Gambar 4.11** Program Menampilkan Tulisan pada LCD 20x4



**Gambar 4.12** Pengujian LCD 20x4

#### 4.8 Pengujian RTC DS1307

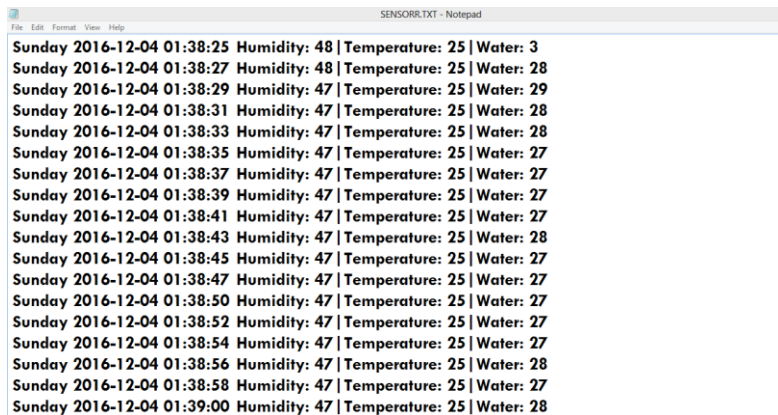
Pengujian RTC DS1307 ini dilakukan untuk mengetahui format tanggal dan waktu sesuai dengan kondisi *real-time*. Pengujian ini dilakukan dengan menggunakan mikrokontroler Arduino dengan menuliskan program seperti pada Gambar 3.20. RTC memerlukan *supply* +5V yang didapatkan dari Arduino, pin SDA ke SDA Arduino dan pin SCL ke SCL Arduino. Berikut merupakan hasil pengujian RTC DS1307 pada Gambar 4.13 :



**Gambar 4.13** Pengujian RTC DS1307

#### 4.9 Pengujian Modul SD Card

Pengujian Modul SD Card ini dilakukan untuk mengetahui kerja dari SD Card yang difungsikan untuk menyimpan nilai *set point* dan pembuatan *datalogger*. Pengujian ini dilakukan dengan menuliskan hari, tanggal dan waktu secara *real-time* serta temperatur dan kelembaban yang disimpan dalam format *txt* dan dengan menu-liskan program seperti pada Gambar 3.21. Pengujian ini menghasilkan seperti pada Gambar 4.14 :



**Gambar 4.14** Pengujian Modul SD Card sebagai Datalogger

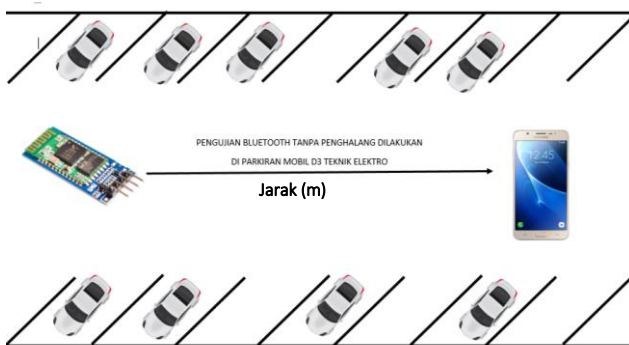
#### 4.10 Pengujian Modul Bluetooth HC-05

Pengujian modul *bluetooth* dilakukan dengan dua pengujian utama, yakni tanpa penghalang, dan dengan penghalang berupa di-

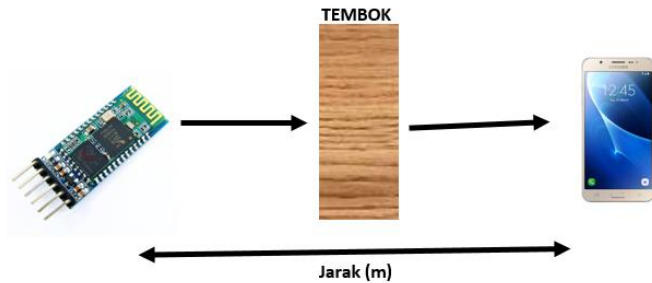
ding. Untuk pengujian tanpa penghalang dilakukan pada tempat lapang dimana tidak ada sesuatu apapun yang menghalangi. sedangkan untuk pengujian dengan penghalang, dilakukan pada suatu ruangan menuju ke ruangan yang lain (terhalang oleh dinding ruangan) seperti pada Gambar 4.16 dan Gambar 4.17. Pada setiap pengujian yang dilakukan, penulis juga mengubah *serial parameter* berupa *baud rate*, *stop bit* dan *parity* mengacu pada metode pengaturan AT Command pada Gambar 3.14. Selanjutnya, menuliskan program seperti pada Gambar 4.15 berikut:

```
char data = 0;
int a;
void setup()
{
    Serial3.begin(38400); //Sets the baud for serial data
}
void loop()
{
    a++;
    Serial3.println(a);
    delay(500);
}
```

**Gambar 4.15** Program Pengujian *Bluetooth*



**Gambar 4.16** Ilustrasi Pengujian Tanpa Penghalang

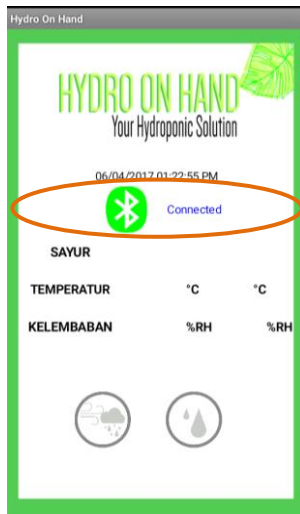


**Gambar 4.17** Ilustrasi Pengujian Dengan Penghalang

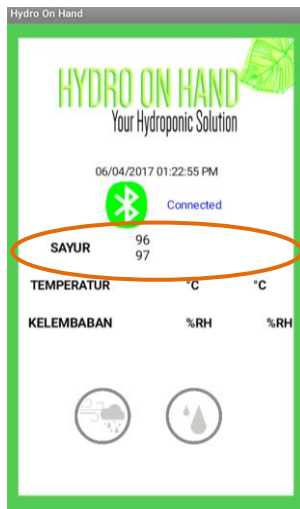
Apabila telah melakukan pengaturan pada setiap *serial parameter* yang akan diuji, maka pengujian dapat dilakukan. Pada tahap pengujian dilakukan dengan cara menghubungkan dua perangkat *bluetooth* yakni HC-05 yang terhubung dengan laptop dengan perangkat *bluetooth* pada *smartphone*. Dimana, jika pada kedua perangkat telah terhubung, maka laptop akan mengirimkan data berupa data berupa angka yang berawal dari 0 hingga 1023 ke tampilan aplikasi yang telah dibuat untuk *smartphone Android*. Selanjutnya, laptop dijauhkan dari *smartphone* hingga jarak maksimal yang dapat dijangkau oleh kedua perangkat.

Dengan demikian, dapat diketahui jarak maksimal dengan cara mengukur jarak dari satu perangkat *bluetooth* HC-05 dengan perangkat *bluetooth* lain.

Pengujian ini bertujuan untuk mengetahui perbedaan jangkauan jarak antara tanpa penghalang atau dengan penghalang. Sebelum melakukan pengujian, pastikan bahwa modul *bluetooth* HC-05 dalam kondisi *slave*. Jika perangkat *Android* dengan *bluetooth* dalam keadaan terhubung maka pada perangkat *Android* terlihat seperti Gambar 4.18. Ketika sudah terhubung selanjutnya adalah proses pengiriman data terkirim atau tidak. Jika terkirim tampilan pada *Android* terlihat seperti Gambar 4.19.



**Gambar 4.18** Tampilan Aplikasi Ketika Sudah Terhubung



**Gambar 4.19** Tampilan Aplikasi Ketika Mengirim Data Berupa Angka

#### 4.10.1 Pengujian *Bluetooth* Tanpa Penghalang

Pengujian jangkauan *bluetooth* yang pertama dilakukan tanpa penghalang antara kedua perangkat *bluetooth* yang bekerja. Pengujian ini dilakukan dengan mengirim data yang ada pada alat untuk ditampilkan pada *smartphone Android*. Pengujian dilakukan dengan mengacu pada Gambar 4.16 pengujian dilakukan dengan mengubah *serial parameter* berupa *baud rate*, *stop bit*, dan *parity*-nya pada setiap pengujian yang akan dilakukan. Tabel 4.10 sampai dengan Tabel 4.15 merupakan data pengujian *bluetooth* tanpa penghalang.

**Tabel 4.10** Pengujian *Bluetooth* Tanpa Penghalang (*Data Bit 8, Stop Bit=1, Parity 0 (None)*)

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Minggu, 4 Juni 2017	2400	5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		25	Terhubung	Terkirim
		50	Terhubung	Terkirim
		80	Terhubung	Terkirim
		95	Terhubung	Terkirim
		100	Tidak Terhubung	Tidak Terkirim
	4800	5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		35	Terhubung	Terkirim
		45	Terhubung	Terkirim
		48	Terhubung	Terkirim
		50	Tidak Terhubung	Tidak Terkirim
	9600	5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		15	Terhubung	Terkirim
		20	Terhubung	Terkirim
		25	Terhubung	Terkirim
		30	Terhubung	Terkirim
		35	Terhubung	Terkirim
		40	Tidak Terhubung	Tidak Terkirim
	38400	5	Terhubung	Terkirim
		10	Terhubung	Terkirim

<b>Waktu</b>	<b>Baud Rate</b>	<b>Jarak (m)</b>	<b>Hasil</b>	<b>Pengiriman Data</b>
Minggu, 4 Juni 2017	38400	35	Terhubung	Terkirim
		50	Terhubung	Terkirim
		60	Terhubung	Terkirim
		65	Terhubung	Terkirim
		68	Terhubung	Terkirim
		70	Tidak Terhubung	Tidak Terkirim

**Tabel 4.11** Pengujian *Bluetooth* Tanpa Penghalang (*Data Bit 8, Stop Bit=1, Parity 1 (Odd)*)

<b>Waktu</b>	<b>Baud Rate</b>	<b>Jarak (m)</b>	<b>Hasil</b>	<b>Pengiriman Data</b>
Minggu, 4 Juni 2017	2400	5	Terhubung	Tidak Terkirim
		10	Terhubung	
		15	Terhubung	
		20	Tidak Terhubung	
	4800	5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		17	Terhubung	
		24	Terhubung	
		30	Tidak Terhubung	
	9600	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		20	Terhubung	
		25	Tidak Terhubung	
	38400	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		16	Terhubung	
		20	Tidak Terhubung	



**Tabel 4.12** Pengujian *Bluetooth* Tanpa Penghalang (*Data Bit 8, Stop Bit=1, Parity 2 (Even)*)

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Minggu, 4 Juni 2017	2400	5	Terhubung	Tidak Terkirim
		10	Terhubung	
		15	Terhubung	
		17	Terhubung	
		19	Terhubung	
		20	Tidak Terhubung	
	4800	3	Terhubung	Tidak Terkirim
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		20	Terhubung	
		28	Terhubung	
		30	Tidak Terhubung	
	9600	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		20	Terhubung	
		26	Terhubung	
		30	Tidak Terhubung	
	38400	3	Terhubung	
		5	Terhubung	
		15	Terhubung	
		30	Terhubung	
		35	Tidak Terhubung	

Pada Tabel 4.10 merupakan data jarak dan respon penerimaan data oleh *bluetooth* pada kondisi tanpa penghalang dan modul *bluetooth* diatur dengan *stop bit=1* dan *parity=0*, jarak paling jauh yang dapat dijangkau yaitu 95 meter pada *baud rate* 2400. Pada Tabel 4.11 modul *bluetooth* diatur dengan *stop bit=1* dan *parity=1*, dan pada Tabel 4.12 modul *bluetooth* diatur dengan *stop bit=1* dan *parity=2*, namun pada kedua pengaturan ini, data tidak dapat diterima

oleh *android* sehingga untuk pengaturan *parity* 1 dan 2 tidak dapat digunakan untuk komunikasi data.

**Tabel 4.13** Pengujian *Bluetooth* Tanpa Penghalang (*Data Bit* 8, *Stop Bit*=2, *Parity* 0 (*None*))

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Minggu 4 Juni 2017	2400	3	Terhubung	Terkirim
		5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		20	Terhubung	Terkirim
		25	Terhubung	Terkirim
		35	Terhubung	Terkirim
		40	Tidak Terhubung	Tidak Terkirim
	4800	5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		25	Terhubung	Terkirim
		35	Terhubung	Terkirim
		50	Terhubung	Terkirim
		60	Terhubung	Terkirim
		80	Terhubung	Terkirim
		90	Terhubung	Terkirim
		105	Tidak Terhubung	Tidak Terkirim
	9600	5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		25	Terhubung	Terkirim
		35	Terhubung	Terkirim
		50	Terhubung	Terkirim
		60	Terhubung	Terkirim
		70	Terhubung	Terkirim
		75	Tidak Terhubung	Tidak Terkirim
	38400	5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		25	Terhubung	Terkirim
		35	Terhubung	Terkirim
		40	Terhubung	Terkirim

**Tabel 4.14** Pengujian *Bluetooth* Tanpa Penghalang (*Data Bit 8, Stop Bit=2, Parity 1 (Odd)*)

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Minggu, 4 Juni 2017	2400	3	Terhubung	Tidak Terkirim
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		26	Terhubung	
		30	Tidak Terhubung	
	4800	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		17	Tidak Terhubung	
	9600	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		21	Terhubung	
		25	Tidak Terhubung	
	38400	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		18	Terhubung	
		19	Terhubung	
		20	Tidak Terhubung	

**Tabel 4.15** Pengujian *Bluetooth* Tanpa Penghalang (*Data Bit 8, Stop Bit=2, Parity 2 (Even)*)

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Minggu, 4 Juni 2017	2400	3	Terhubung	Tidak Terkirim
		5	Terhubung	
		10	Terhubung	

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Minggu, 4 Juni 2017	2400	12	Terhubung	Tidak Terkirim
		15	Tidak Terhubung	
	4800	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		16	Terhubung	
		18	Tidak Terhubung	
	9600	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		20	Terhubung	
		25	Tidak Terhubung	
	38400	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		16	Terhubung	
		25	Terhubung	
		30	Tidak Terhubung	

Pada Tabel 4.13 merupakan data jarak dan respon penerimaan data oleh *bluetooth* pada kondisi tanpa penghalang dan modul *bluetooth* diatur dengan *stop bit*=2 dan *parity*=0, jarak paling jauh yang dapat dijangkau yaitu kurang lebih 100 meter pada *baud rate* 4800. Pada Tabel 4.14 modul *bluetooth* diatur dengan *stop bit*=2 dan *parity*=1, dan pada Tabel 4.15 modul *bluetooth* diatur dengan *stop bit*=2 dan *parity*=2, namun pada kedua pengaturan ini, data tidak dapat diterima oleh *android* sehingga untuk pengaturan *parity* 1 dan 2 tidak dapat digunakan untuk komunikasi data.

#### 4.10.2 Pengujian Bluetooth Dengan Penghalang

Pengujian jangkauan *bluetooth* yang kedua dilakukan dengan penghalang berupa dinding antara kedua perangkat *bluetooth* yang bekerja. Pengujian ini dilakukan dengan cara mengacu pada Gambar 4.17.

Pengujian ini menghasilkan data pada Tabel 4.16 sampai dengan Tabel 4.21 sebagai berikut :

**Tabel 4.16** Pengujian *Bluetooth* Dengan Penghalang (*Data Bit 8, Stop Bit=1, Parity 0 (None)*)

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Kamis, 1 Juni 2017	2400	3	Terhubung	Terkirim
		5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		15	Terhubung	Terkirim
		20	Terhubung	Terkirim
		21	Terhubung	Terkirim
		25	Tidak Terhubung	Tidak Terkirim
	4800	3	Terhubung	Terkirim
		5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		15	Terhubung	Terkirim
		20	Terhubung	Terkirim
		25	Terhubung	Terkirim
		30	Tidak Terhubung	Tidak Terkirim
	9600	3	Terhubung	Terkirim
		5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		15	Terhubung	Terkirim
		20	Terhubung	Terkirim
		25	Terhubung	Terkirim
		30	Terhubung	Terkirim
		35	Tidak Terhubung	Tidak Terkirim
	38400	3	Terhubung	Terkirim
		5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		15	Terhubung	Terkirim
		20	Terhubung	Terkirim
		23	Terhubung	Terkirim
		25	Tidak Terhubung	Tidak Terkirim

**Tabel 4.17** Pengujian *Bluetooth* Dengan Penghalang (*Data Bit 8, Stop Bit=1, Parity 1 (Odd)*)

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Kamis, 1 Juni 2017	2400	3	Terhubung	Tidak Terkirim
		5	Terhubung	
		10	Terhubung	
		11	Tidak Terhubung	
	4800	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
	9600	11	Tidak Terhubung	
		3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		11	Tidak Terhubung	
	38400	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		11	Tidak Terhubung	

**Tabel 4.18** Pengujian *Bluetooth* Dengan Penghalang (*Data Bit 8, Stop Bit=1, Parity 2 (Even)*)

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Kamis, 1 Juni 2017	2400	3	Terhubung	Tidak Terkirim
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		20	Tidak Terhubung	
	4800	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		17	Terhubung	
		18	Terhubung	
		19	Terhubung	
		20	Tidak Terhubung	

Waktu	Baud Rate	Jarak (m)		Pengiriman Data
Kamis, 1 Juni 2017	9600	3	Terhubung	Tidak Terkirim
		5	Terhubung	
		10	Terhubung	
		17	Terhubung	
		20	Tidak Terhubung	
	38400	3	Terhubung	
		4	Terhubung	
		5	Terhubung	
		10	Terhubung	
		12	Terhubung	
		15	Terhubung	
		16	Terhubung	
		20	Tidak Terhubung	

Pada Tabel 4.16 merupakan data jarak dan respon penerimaan data oleh *bluetooth* pada kondisi dengan penghalang dan modul *bluetooth* diatur dengan *stop bit=1* dan *parity=0*, jarak paling jauh yang dapat dijangkau yaitu kurang lebih 30 meter pada *baud rate* 9600. Pada Tabel 4.17 modul *bluetooth* diatur dengan *stop bit=1* dan *parity=1*, dan pada Tabel 4.18 modul *bluetooth* diatur dengan *stop bit=1* dan *parity=2*, namun pada kedua pengaturan ini, data tidak dapat diterima oleh *android* sehingga untuk pengaturan *parity* 1 dan 2 tidak dapat digunakan untuk komunikasi data.

**Tabel 4.19** Pengujian *Bluetooth* Dengan Penghalang (Data Bit 8, Stop Bit=2, Parity 0 (None))

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Kamis 1 Juni 2017	2400	3	Terhubung	Terkirim
		5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		15	Terhubung	Terkirim
		20	Terhubung	Terkirim
		25	Terhubung	Terkirim
		28	Terhubung	Terkirim

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Kamis, 1 Juni 2017	4800	3	Terhubung	Terkirim
		5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		15	Terhubung	Terkirim
		20	Terhubung	Terkirim
		25	Tidak Terhubung	Tidak Terkirim
	9600	3	Terhubung	Terkirim
		5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		15	Terhubung	Terkirim
		19	Terhubung	Terkirim
		20	Tidak Terhubung	Tidak Terkirim
	38400	3	Terhubung	Terkirim
		5	Terhubung	Terkirim
		10	Terhubung	Terkirim
		15	Terhubung	Terkirim
		20	Terhubung	Terkirim
		25	Terhubung	Terkirim
		27	Terhubung	Terkirim
		30	Tidak Terhubung	Tidak Terkirim

**Tabel 4.20** Pengujian *Bluetooth* Dengan Penghalang (*Data Bit 8, Stop Bit=2, Parity 1 (Odd)*)

Waktu	Baud Rate	Jarak (m)	Hasil	Respon Pengiriman Data
Kamis, 1 Juni 2017	2400	3	Terhubung	Tidak Terkirim
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		20	Tidak Terhubung	
	4800	3	Terhubung	
		5	Terhubung	
		10	Terhubung	



Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Kamis, 1 Juni 2017	4800	15	Terhubung	Tidak Terkirim
		20	Tidak Terhubung	
	9600	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		15	Terhubung	
		20	Tidak Terhubung	
	38400	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		15	Tidak Terhubung	

**Tabel 4.21** Pengujian *Bluetooth* Dengan Penghalang (*Data Bit 8, Stop Bit=2, Parity 2 (Even)*)

Waktu	Baud Rate	Jarak (m)	Hasil	Pengiriman Data
Kamis, 1 Juni 2017	2400	5	Terhubung	Tidak Terkirim
		10	Terhubung	
		17	Terhubung	
		20	Tidak Terhubung	
	4800	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		16	Terhubung	
		20	Tidak Terhubung	
	9600	5	Terhubung	
		10	Terhubung	
		16	Terhubung	
		20	Tidak Terhubung	
	38400	3	Terhubung	
		5	Terhubung	
		10	Terhubung	
		17	Terhubung	
		20	Tidak Terhubung	

Pada Tabel 4.19 merupakan data jarak dan respon penerimaan data oleh *bluetooth* pada kondisi dengan penghalang dan modul *bluetooth* diatur dengan *stop bit*=2 dan *parity*=0, jarak paling jauh yang dapat dijangkau yaitu kurang lebih 28 meter pada *baud rate* 2400. Pada Tabel 4.20 modul *bluetooth* diatur dengan *stop bit*=2 dan *parity*=1, dan pada Tabel 4.21 modul *bluetooth* diatur dengan *stop bit*=2 dan *parity*=2, namun pada kedua pengaturan ini, data tidak dapat diterima oleh *android* sehingga untuk pengaturan *parity* 1 dan 2 tidak dapat digunakan untuk komunikasi data.

#### 4.11 Pengujian Aplikasi Android

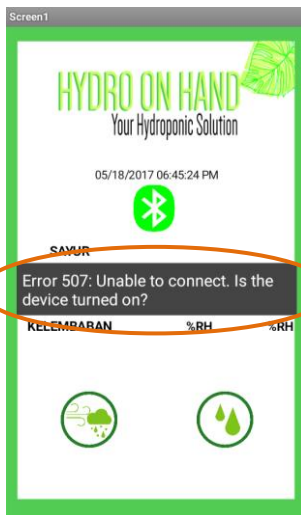
Pada pengujian aplikasi *android* ini bertujuan untuk mengetahui apakah aplikasi yang telah dibuat dapat menerima data dari alat dengan baik atau tidak. Pengujian ini dilakukan dengan menguji sistem secara keseluruhan serta menggunakan *smartphone android* dengan versi 5.1.1 (versi *Lollipop*) dan program yang digunakan mengacu pada Gambar 3.25 dan 3.26.

Pengujian koneksi antara sistem *android* dan sistem alat keseluruhan dilakukan dengan cara mengkoneksikan *bluetooth* antar dua perangkat yang bekerja. Jika sistem pada alat telah sudah terkoneksi dengan *smartphone android*, maka pada tampilan aplikasi akan tampak seperti Gambar 4.18. Namun, jika sistem pada alat tidak terkoneksi dengan *smartphone* maka pada tampilan aplikasi akan muncul pemberitahuan bahwa kedua perangkat tidak terhubung (*error*) seperti pada Gambar 4.20.

Selanjutnya, apabila perangkat *bluetooth* pada *smartphone Android* sudah terhubung dengan perangkat *bluetooth* pada alat, maka aplikasi akan menampilkan data berupa parameter sayuran yang terdapat pada alat, seperti yang ditunjukkan pada Gambar 4.21.

Telah diketahui pada pengujian perangkat *bluetooth*, bahwa jarak sangat mempengaruhi pengiriman dan penerimaan data oleh kedua perangkat *bluetooth* yang bekerja. Gambar 4.22 merupakan tampilan aplikasi yang sudah tidak dapat melakukan pengiriman dan penerimaan data karena pengaruh jarak maksimal yang dicapai atau dapat pula dikatakan koneksi antar *bluetooth* telah terputus sehingga proses komunikasi data tidak dapat berlangsung.

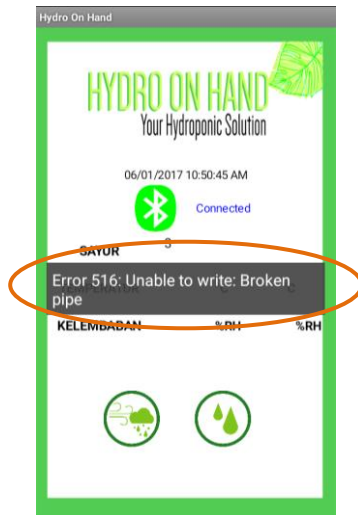
Gambar 4.23 dan Gambar 4.24 merupakan tampilan pada aplikasi *android* yang menunjukkan tentang pengaturan nyala atau tidaknya aktuator yang berupa pompa dan *humidifier*.



**Gambar 4.20** Tampilan Aplikasi Ketika Tidak Terhubung



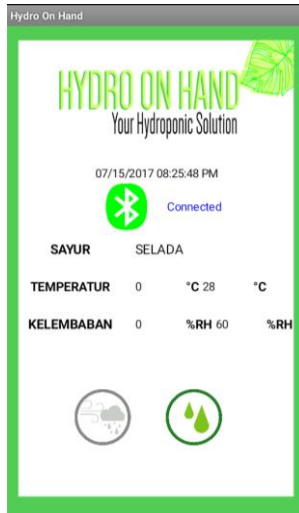
**Gambar 4.21** Tampilan Aplikasi Ketika Menerima Data



**Gambar 4.22** Tampilan Aplikasi Ketika Koneksi Terputus



**Gambar 4.23** Tampilan Aplikasi Ketika *Humidifier* Aktif



**Gambar 4.24** Tampilan Aplikasi Ketika Pompa Aktif

#### 4.12 Pengujian Keseluruhan Sistem Alat

Pengujian keseluruhan sistem alat dilakukan dengan cara pengimplementasian langsung pada lingkungan. Pengujian sistem ini dilakukan setelah tahap pengujian sensor dan modul lainnya. Pengujian sistem ini dilakukan untuk mengetahui dan mengukur tingkat keakuratan dari serangkaian modul yang saling terintegrasi. Tujuannya, yaitu untuk membuktikan perancangan sistem telah berjalan dengan baik dan mengetahui hasil dari tujuan pembuatan sistem telah tercapai atau tidak.

Sebelum melakukan pengujian sistem pada alat, dilakukan penanaman benih sayuran pada *starter kit*. Sayuran yang akan diuji yaitu selada dengan *setting point* temperatur 27°C dan kelembaban 80%. Dibutuhkan waktu sekitar dua hari untuk pemecahan benih sayuran. Penghitungan hari dilakukan setelah pecah benih yaitu di mulai pada hari Sabtu, 8 Juli 2017 seperti yang ditunjukkan Gambar 4.25.

Pelaksanaan pengujian di lakukan selama 15 menit pada 3 waktu berbeda yaitu pagi, siang, dan sore, dimana pada waktu-waktu tersebut mengalami perubahan temperatur yang dapat mengancam kelangsungan hidup benih sayuran yang telah ditanam. Data hasil pengujian terdapat pada Tabel 4.22 sampai dengan Tabel 4.24.



**Gambar 4.25** Penyemaian Benih Selada Hari ke-1 (Sabtu, 8 Juli 2017)

**Tabel 4.22** Pengujian Sistem pada Pukul 09.00-09.15

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
29°C	61%	Nyala	2,7 cm	Mati
29°C	61%	Nyala	2,6 cm	Mati
28°C	62%	Nyala	2,0 cm	Nyala
28°C	62%	Nyala	1,7 cm	Nyala
27°C	63%	Nyala	1,6 cm	Nyala

**Tabel 4.23** Pengujian Sistem pada Pukul 12.30-12.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
30°C	58%	Nyala	2,7 cm	Mati
30°C	57%	Nyala	2,6 cm	Mati
29°C	61%	Nyala	1,9 cm	Nyala
28°C	60%	Nyala	1,4 cm	Nyala
29°C	61%	Nyala	1,03 cm	Nyala

**Tabel 4.24** Pengujian Sistem pada Pukul 16.30-16.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
29°C	59%	Nyala	2,57 cm	Mati
28°C	59%	Nyala	2,43 cm	Nyala
28°C	61%	Nyala	2,0 cm	Nyala
27°C	62%	Nyala	2,3 cm	Nyala
27°C	64%	Nyala	1,75 cm	Nyala

Proses penyemaian hari pertama pada Sabtu, 8 Juli 2017 terlihat pada Gambar 4.25 bahwa tanaman mulai pecah benih. pengambilan data dimulai pada pagi hari pukul 09.00-09.15 diawali dengan mengisi air pada bak *starter kit* hingga 2,7 cm. Saat sistem dinyalakan, temperatur menunjukkan melebihi *setting point* yaitu 29°C, maka *humidifier* menyala sehingga temperatur dapat mencapai 27°C. Namun, untuk kelembaban rata-rata bernilai 60% kurang dari 80%. Pada pengujian selanjutnya pada pukul 12.30-12.45 dan pengujian pada pukul 16.30-16.45 dilakukan pengisian ulang untuk air pada bak *starter kit*. Untuk pengujian sampai dengan hari ke 6 terdapat pada Lampiran C.

Setelah dilakukan pengujian selama enam hari terhadap sembi-lan benih selada untuk tahap penyemaian, terdapat delapan benih yang dapat tumbuh dengan baik, namun satu benih lainnya tidak dapat tumbuh dan kering. Dengan demikian, presentase keberhasilan dari alat ini dapat dihitung dengan menggunakan Persamaan 4.6 sebagai berikut :

$$\% \text{Keberhasilan} = \frac{\text{JumlahBenihTumbuh}}{\text{JumlahBenihTotal}} \times 100\% \dots\dots\dots(4.6)$$

$$\% \text{Keberhasilan} = \frac{8}{9} \times 100\%$$

$$\% \text{Keberhasilan} = 88,89 \%$$

-----Halaman ini sengaja dikosongkan-----



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Pada penelitian kali ini, dapat diambil kesimpulan antara lain sebagai berikut:

1. Serial parameter komunikasi *bluetooth* yang dapat digunakan untuk jarak yang paling jauh yaitu :
  - a. 2400, 0, 0 dengan jarak maksimal 90 m (tanpa penghalang)
  - b. 9600, 0, 0 dengan jarak maksimal 30m (dengan penghalang)
2. Pada pengaturan *parity* 1 dan 2 tidak dapat digunakan untuk komunikasi data karena pada pengaturan tersebut proses pengiriman data tidak dapat berlangsung.
3. Saat dilakukan pengujian sistem alat terdapat kesalahan (*error*) dalam hal penerimaan data yang dikirim ke *smartphone android* sehingga perlu dilakukan pengkoneksian (*pairing*) ulang antara kedua perangkat *bluetooth*. Selain itu, selama 15 menit (900 detik) pengujian alat ditemukan kesalahan tampilan (*error display*) berupa pertukaran data sebanyak 20 kali antar parameter yang ditampilkan pada *smartphone android*.
4. Persentase *error* pada sensor HSM-20G relatif kecil yaitu pada kelembaban sebesar 0% dan temperatur sebesar 0,7% maka dari itu sensor HSM-20G ini dapat digunakan pada sistem ini.
5. Persentase *error* pada sensor *Water Level* Funduino, pada ketinggian 0,5 sampai dengan 2 cm, % *error* sangat tinggi. Namun pada ketinggian 2,5 sampai dengan 3 cm, % *error* terlampau kecil yaitu hanya sekitar 0,4 – 1,3 %. Pada alat ini yang digunakan adalah ketinggian 2,5 sampai dengan 3 cm karena menyesuaikan dengan kebutuhan tanaman sehingga sensor ini dapat digunakan pada alat ini.
6. Pada pengujian sistem keseluruhan, dilakukan penanaman 9 buah benih selada. Namun, saat pengimplementasian dengan alat, terdapat 1 buah benih yang gagal tumbuh. Sehingga presentase keberhasilan dari sistem alat yaitu 88,89 persen.

## 5.2 **Saran**

Saran untuk penelitian selanjutnya yaitu diharapkan dapat menggunakan perangkat komunikasi data selain *bluetooth* yang dapat menjangkau jarak yang lebih jauh. Selain itu, perlu adanya inovasi dari aplikasi *Android* yang telah dibuat salah satunya dapat ditambahkan indikator yang menyatakan status dari aktuator apakah menyala atau tidak. Serta juga untuk lebih memerhatikan intensitas cahaya yang dibutuhkan tanaman ketika ditanam di luar atau di dalam. Diharapkan alat ini juga dapat dikembangkan menjadi alat untuk media tanam hidroponik secara otomatis yang dapat ditanam didalam rumah atau yang sering dikenal *Grow Box*.

## DAFTAR PUSTAKA

- [1] Roberto, Keith, *How To Hydroponic*, Future Garden, New York, 2000.
- [2] Hanum, Chairani, *Teknik Budidaya Tanaman*, BSE, Jakarta, 2003.
- [3] ....., “Perkembangan dan Pertumbuhan pada Pola Cocok Tanam Selada”, *Jurnal Fakultas Pertanian*, Universitas Lampung, 2013.
- [4] ....., “*Dasar dan Teori Komunikasi Data*” , [http://elib.unikom.ac.id/files/disk1/400/jbptunikompp-gdl-durahmanni-199 82-7-babii.p df](http://elib.unikom.ac.id/files/disk1/400/jbptunikompp-gdl-durahmanni-199%2082-7-babii.p%20df). diakses pada 08 April 2017
- [5] Christofer, Gerry., Sujaini, Herry., Irwansyah, M.Azhar., “Rancang Bangun Aplikasi Early Warning Dengan Pemanfaatan Pengukuran Suhu Ruangan Berbasis Arduino Mega 2560”, *Tugas Akhir*, Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Tanjungpura, Tanjungpura, 2014.
- [6] Latino, Nico., Hamzah, Amir., “Perbaikan Faktor Daya dengan Implementasi TRIAC Berbasis Mikrokontroler”, *Jurnal Teknik Elektro*, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Riau, Riau, 2016.
- [7] Saputra, Indra., Triyanto, Dedi., Ruslianto, Ikhwan., “Sistem Kendali Temperatur, Kelembaban, dan Level Air pada Pertanian Pola Hidroponik”, *Jurnal Elektronika*, Jurusan Teknik Elektronika, Fakultas Teknik, Universitas Tanjungpura, Pontianak, 2015.
- [8] Putranto, Tommy Dwi., dan Rochman, Bayu Fachtur. , “Rancang Bangun Sistem Otomasi Pemberian Nutrisi dan Pencahayaan untuk Tahap Penyemaian Benih Selada pada Pekebunan Surabaya Hidroponik”, *Tugas Akhir*, Jurusan D3 Teknik Elektro, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.
- [9] Santi, Karina Eka., Ramadhani, Erwin., “AMC (*Automatic Money Changer*)”, *Tugas Akhir*, Jurusan D3 Teknik Elektro, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.
- [10] Holmes, *How Does an Ultrasonic Humidifier Work*, <http://www.holmesproducts.com/blog/archive/2014/october/ho>

w-does-an-ultrasonic-humidifier-work%3F.html, diakses pada 20 Juni 2017.

- [11] Yenny, Ervina, “Rancang Bangun Sistem Penyiraman Sayur Sawi (*Brassica chinensi L.*) Menggunakan Sensor Kelembaban dan Sensor Intensitas Cahaya Berbasis *Fuzzy Logic*”, **Tugas Akhir**, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember, Jember, 2015.
- [12] ....., “**Dasar Komunikasi Data Bluetooth-Politeknik Negeri Sriwijaya**,” <http://eprints.polsri.ac.id/1873/7/bab%202.pdf>. diakses pada 08 April 2017
- [13] ....., “**App Inventor**”, [https://id.wikipedia.org/wiki/App\\_Inventor](https://id.wikipedia.org/wiki/App_Inventor). diakses pada 28 April 2017.
- [14] Rahmawati, Rizqi., Dien Muhammad Faris, “Perancangan Prototipe Mesin Pemotong Rumput Taman yang Dikendalikan Menggunakan *Smartphone* Via *Bluetooth*”, **Tugas Akhir**, Jurusan D3 Teknik Elektro, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.

## LAMPIRAN A

### A.1. Program Arduino

```
#include <LiquidCrystal.h>
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include "RTClib.h"
char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed",
"Thu", "Fri", "Sat"};
RTC_DS1307 RTC;
LiquidCrystal lcd(10, 9, 36, 38, 6, 5);
int param=0;
int button_up,button_down,sel,metu=0;
char tampil[20];
int kolom=0;
int baris=0;
int temp;
int suhuS, suhuS1;
int lembabS, lembabS1;
int air, air1;
int state;
int lihat;
const int chipSelect = 4;
const int sensorw = A0;
float val;
float h;
const int pinPompa = 48;
const int pinHumi = 49;
byte serialA;
float value1;
int humidity;
float volth;
int temperature;
float value2;
char data = 0;
int mot = 11;
int in=40;
```

```

int in1=42;
int kondisi;
boolean f=true;
boolean f1=true;
int f2=0;
int ldr;
byte suhuu[8] = {
    0b00100,
    0b01010,
    0b01010,
    0b01010,
    0b01010,
    0b01010,
    0b10001,
    0b10001,
    0b01110
};
byte banyu[8] = {
    0b00000,
    0b00100,
    0b01010,
    0b10001,
    0b10001,
    0b10001,
    0b01110,
    0b00000
};
byte anyeb[8] = {
    0b11100,
    0b10100,
    0b11000,
    0b10100,
    0b00000,
    0b00101,
    0b00111,
    0b00101
};
void setup()
{
    pinMode(2,INPUT_PULLUP);

```

```

pinMode(3,INPUT_PULLUP);
pinMode(4,INPUT_PULLUP);
pinMode(12,INPUT_PULLUP);
pinMode(pinPompa,OUTPUT);
pinMode(pinHumi,OUTPUT);
pinMode(11, OUTPUT);
pinMode(40, OUTPUT);
pinMode(42, OUTPUT);
// pinMode(49, OUTPUT);
lcd.begin(20, 4);
lcd.createChar(0, suhuu);
lcd.createChar(1, anyeb);
lcd.createChar(2, banyu);
Serial3.begin(9600);
//Serial.flush();
Wire.begin();
RTC.begin();
Serial.print("Initializing SD card...");
if (!SD.begin(53)) {
  Serial.println("Card failed, or not present");
  return;
}
Serial.println("card initialized.");
if (! RTC.isrunning()) {
  Serial.println("RTC is NOT running!");
  // following line sets the RTC to the date & time this sketch was
  compiled
  RTC.adjust(DateTime(__DATE__, __TIME__));
}
}

void water (){
  val = analogRead (sensorw);
  float volts = ((val*4.4)/1023);
  h=(volts*1.218905053)-0.565117967;
  delay (1000);
}

void motor(){
  ldr=analogRead(4);
  if (ldr>600&&f==true)

```

```

{
  analogWrite(mot, 255);
  digitalWrite(in, LOW);
  digitalWrite(in1, HIGH);
  delay(2000);
  analogWrite(mot, 0);
  f=false;
  f1=true;
}
if (ldr<600&&f1==true)
{
  analogWrite(mot, 255);
  digitalWrite(in, HIGH);
  digitalWrite(in1, LOW);
  delay(2000);
  analogWrite(mot, 0);
  f=true;
  f1=false;
}
}
void bt(){
  if (state==0){
    Serial3.print("SELADA");
    Serial3.print("|");
    Serial3.print(suhuS1);
    Serial3.print("|");
    Serial3.print(temperature);
    Serial3.print("|");
    Serial3.print(lembabS1);
    Serial3.print("|");
    Serial3.print(humidity);
  }
  else if(state==1){
    Serial3.print("PAK COY");
    Serial3.print("|");
    Serial3.print(suhuS1);
    Serial3.print("|");
    Serial3.print(temperature);
    Serial3.print("|");
  }
}

```



```

        Serial3.print(lembabS1);
        Serial3.print("|");
        Serial3.print(humidity);
    }
    else if (state==2){
        Serial3.print("BAYAM");
        Serial3.print("|");
        Serial3.print(suhuS1);
        Serial3.print("|");
        Serial3.print(temperature);
        Serial3.print("|");
        Serial3.print(lembabS1);
        Serial3.print("|");
        Serial3.print(humidity);
    }
    Serial3.println("|");
    if (Serial3.available()>0) {
        serialA = Serial3.read();
        Serial3.println(serialA);
    }
    switch(serialA){
        case 49:
            digitalWrite(pinPompa, HIGH);
            kondisi=1;
            break;
        case 50:
            digitalWrite(pinPompa, LOW);
            break;
        case 51:
            digitalWrite(pinHumi, LOW);
            break;
        case 52:
            digitalWrite(pinHumi, HIGH);
            kondisi=2;
            break;
    }
}
}
void hsm()
{

```

```

    value1 = analogRead(2); //Read data from analog pin and store it to
value variable
    volth = ((value1*4.3)/1023);
    humidity = (((volth*30.8547580362547)-11.5038176856011));
    value2 = analogRead(1); //Read data from analog pin and store it to
value variable
    temperature = (((value2*0.106068859)-37.90680477));
}
void jam()
{
    lcd.clear();
    //RTC.adjust(DateTime(__DATE__, __TIME__));
    DateTime now = RTC.now();
    lcd.setCursor(0,0);
    lcd.print("Hidroponik Surabaya");
    lcd.setCursor(5, 1);
    lcd.print(now.day(), DEC);
    lcd.print('/');
    lcd.print(now.month(), DEC);
    lcd.print('/');
    lcd.print(now.year(), DEC);
    lcd.print(' ');
    lcd.setCursor(5, 2);
    if (now.hour()<10)
    lcd.print('0');
    lcd.print(now.hour(), DEC);
    lcd.print(':');
    if (now.minute()<10)
    lcd.print('0');
    lcd.print(now.minute(), DEC);
    lcd.print(':');
    if (now.second()<10)
    lcd.print('0');
    lcd.print(now.second(), DEC);
    lcd.setCursor(1,3);
    lcd.print(char(0));
    lcd.setCursor(2,3);
    Serial.print(now.day(), DEC);
    Serial.print('/');

```

```

Serial.print(now.month(), DEC);
Serial.print('/');
Serial.print(now.year(), DEC);
Serial.println(' ');
if (now.hour()<10)
Serial.print('0');
Serial.print(now.hour(), DEC);
Serial.print(':');
if (now.minute()<10)
Serial.print('0');
Serial.print(now.minute(), DEC);
Serial.print(':');
if (now.second()<10)
Serial.print('0');
Serial.print(now.second(), DEC);

hsm();
Serial.print(char(0));
lcd.print(temperature);
lcd.setCursor(8,3);
lcd.print(char(1));
lcd.setCursor(9,3);
lcd.print(humidity);

int v=analogRead(A1);
lcd.setCursor(15,3);
lcd.print(char(2));
lcd.setCursor(16,3);
lcd.print(v);
}
void SuhuSelada()
{
  lcd.clear();
  if (button_down==0){
    lcd.setCursor(0,0);
    lcd.print("Suhu=");
    lcd.setCursor(6,0);
    suhuS++;
    if (suhuS==31){

```

```

        suhuS=0;
        suhuS++;
    }
    lcd.print(suhuS);
    suhuS1=suhuS;
    delay(10);
}
if (button_down==1){
    lcd.setCursor(0,0);
    lcd.print("Suhu=");
    lcd.setCursor(6,0);
    lcd.print(suhuS1);
    delay(10);
}
}
void LembabSelada(){
    lcd.clear();
    if (button_down==0){
        lcd.setCursor(0,0);
        lcd.print("Lembab=");
        lcd.setCursor(8,0);
        lembabS++;
        if (lembabS==80){
            lembabS=0;
            lembabS++;
        }
        lcd.print(lembabS);
        lembabS1=lembabS;
        delay(10);
    }
    if (button_down==1){
        lcd.setCursor(0,0);
        lcd.print("Lembab=");
        lcd.setCursor(8,0);
        lcd.print(lembabS1);
        delay(10);
    }
}
void AirSawi(){

```

```

lcd.clear();
if (button_down==0){
    lcd.setCursor(0,0);
    lcd.print("Air=");
    lcd.setCursor(5,0);
    air++;
    if (air==31){
        air=0;
        air++;
    }
    lcd.print(air);
    air1=air;
    delay(10);
}
if (button_down==1){
    lcd.setCursor(0,0);
    lcd.print(" Air=");
    lcd.setCursor(5,0);
    lcd.print(air1);
    delay(10);
}
}

void datalogger(){
    DateTime now = RTC.now();
    File dataFile = SD.open("3.txt", FILE_WRITE);
    if (state==0){
        dataFile.println("SELADA");
    }
    else if(state==1){
        dataFile.println("PAK COY");
    }
    else if (state==2){
        dataFile.println("BAYAM");
    }
    dataFile.println(String (now.day(), DEC)+"/"+ String
(now.month(), DEC)+"/"+String (now.year(), DEC) +" "+ String
(now.hour(), DEC)+":"+ String (now.minute(), DEC)+":"+ String
(now.second(), DEC)+" "+ "\tHumidity: "+ String (humidity)+" "+
+ "\tTemperature: "+ String (temperature));
}

```

```

dataFile.close();
delay(100);
}
void loop()
{
  if(kolom==0&&baris==0){
    lcd.setCursor(5,0);
    lcd.print("Hidroponik");
  }
  if(kolom==1&&baris==0){
    jam();
    datalogger();
    motor();
    if (temperature>suhuS1||humidity>lembabS1){
      digitalWrite(49, LOW);
    }
    else { digitalWrite(49, HIGH);}
    if (sensorw>2.5){
      digitalWrite(48, HIGH);
    }
    else { digitalWrite(48, LOW);}
  }
  if(kolom==2&&baris==0){
    bt();
    if(temp==0){
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print(">Pilih Sayur");
      lcd.setCursor(0,1);
      lcd.print(" Aktifkan Humidifier");
      lcd.setCursor(0,2);
      lcd.print(" Aktifkan Pompa");
      lcd.setCursor(0,3);
      lcd.print(" => Atap");
      lcd.setCursor(11,3);
      lcd.print(" <= Atap");
      delay(100);
    }
    else if(temp==1){

```

```

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Pilih Sayur");
    lcd.setCursor(0,1);
    lcd.print(">Aktifkan Humidifier");
    lcd.setCursor(0,2);
    lcd.print(" Aktifkan Pompa");
    lcd.setCursor(0,3);
    lcd.print(" => Atap");
    lcd.setCursor(11,3);
    lcd.print(" <= Atap");
    delay(100);
}
else if(temp==2){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Pilih Sayur");
    lcd.setCursor(0,1);
    lcd.print(" Aktifkan Humidifier");
    lcd.setCursor(0,2);
    lcd.print(">Aktifkan Pompa");
    lcd.setCursor(0,3);
    lcd.print(" => Atap");
    lcd.setCursor(11,3);
    lcd.print(" <= Atap");
    delay(100);
}
else if(temp==3){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Pilih Sayur");
    lcd.setCursor(0,1);
    lcd.print(" Aktifkan Humidifier");
    lcd.setCursor(0,2);
    lcd.print(" Aktifkan Pompa");
    lcd.setCursor(0,3);
    lcd.print(">=> Atap");
    lcd.setCursor(11,3);
    lcd.print(" <= Atap");
}

```

```

        delay(100);
    }
    else if(temp==4){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" Pilih Sayur");
        lcd.setCursor(0,1);
        lcd.print(" Aktifkan Humidifier");
        lcd.setCursor(0,2);
        lcd.print(" Aktifkan Pompa");
        lcd.setCursor(0,3);
        lcd.print(" => Atap");
        lcd.setCursor(11,3);
        lcd.print("> <= Atap");
        delay(100);
    }
    analogWrite(mot, 0);
}
if(kolom==3&&baris==0){
    bt();
    if(temp==0){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(">Selada");
        lcd.setCursor(0,1);
        lcd.print(" Pak Coy");
        lcd.setCursor(0,2);
        lcd.print(" Bayam");
        delay(100);
        state=0;
    }
    else if(temp==1){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" Selada");
        lcd.setCursor(0,1);
        lcd.print(">Pak Coy");
        lcd.setCursor(0,2);
        lcd.print(" Bayam");
    }
}

```



```

        delay(100);
        state=1;
    }
    else if(temp==2){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" Selada");
        lcd.setCursor(0,1);
        lcd.print(" Pak Coy");
        lcd.setCursor(0,2);
        lcd.print(">Bayam");
        delay(100);
        state=2;
    }
}
if(kolom==3&&baris==1){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("HUMI ON");
    delay(100);
    digitalWrite(49, LOW);
    delay(100);
}
else {digitalWrite(49,HIGH);}
if(kolom==3&&baris==2){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("pompa ON");
    digitalWrite(48, HIGH);
    delay(100);
}
else {digitalWrite(48,LOW);}
if(kolom==3&&baris==3){
    f2=1;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("TUTUP");
}
if(kolom==3&&baris==4){

```

```

    f2=2;
    lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("BUKA");
    }
    if(kolom==4&&baris==0){
        bt();
        if(temp==0){
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(">Isi/Edit Set Poin");
            lcd.setCursor(0,1);
            lcd.print(" Lihat Set Poin");
            delay(100);
        }
        else if(temp==1){
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(" Isi/Edit Set Poin");
            lcd.setCursor(0,1);
            lcd.print(">Lihat Set Poin");
            delay(100);
            state=3;
        }
    }
}
if(kolom==4&&baris==1){
    if(temp==0){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(">Isi/Edit Set Poin");
        lcd.setCursor(0,1);
        lcd.print(" Lihat Set Poin");
        delay(100);
    }
    else if(temp==1){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" Isi/Edit Set Poin");
        lcd.setCursor(0,1);
    }
}

```

```

        lcd.print(">Lihat Set Poin");
        delay(100);
        state=4;
    }
}
if(kolom==4&&baris==2){
    if(temp==0){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(">Isi/Edit Set Poin");
        lcd.setCursor(0,1);
        lcd.print(" Lihat SetPoin");
        delay(100);
    }
    else if(temp==1){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" Isi/Edit Set Poin");
        lcd.setCursor(0,1);
        lcd.print(">Lihat Set Poin");
        delay(100);
        state=5;
    }
}
if(kolom==5&&baris==0){
    bt();
    if(temp==0){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(">Suhu");
        lcd.setCursor(0,1);
        lcd.print(" Lembab");
        delay(100);
    }
    else if(temp==1){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(" Suhu");
        lcd.setCursor(0,1);
    }
}

```

```

        lcd.print(">Lembab");
        delay(100);
    }
}
if(kolom==6&&baris==0){
    bt();
    if (state==0){
        SuhuSelada();
        lcd.setCursor(0,2);
        lcd.print("Suhu Selada");
        lcd.setCursor(0,3);
        lcd.print("Tekan Tombol UP");
    }
    else if (state==1){
        SuhuSelada();
        lcd.setCursor(0,2);
        lcd.print("Suhu Pak Coy");
        lcd.setCursor(0,3);
        lcd.print("Tekan Tombol UP");
    }
    else if (state==2){
        SuhuSelada();
        lcd.setCursor(0,2);
        lcd.print("Suhu Bayam");
        lcd.setCursor(0,3);
        lcd.print("Tekan Tombol UP");
    }
    lcd.setCursor(0,1);
    lcd.print("Suhu Sekarang=");
    lcd.print(temperature);
}
if(kolom==6&&baris==1){
    LembabSelada();
    if (state==0){
        lcd.setCursor(0,2);
        lcd.print("Lembab Selada");
        lcd.setCursor(0,3);
        lcd.print("Tekan Tombol UP");
    }
}

```

```

else if (state==1){
    LembabSelada();
    lcd.setCursor(0,2);
    lcd.print("Lembab Pak Coy");
    lcd.setCursor(0,3);
    lcd.print("Tekan Tombol UP");
}
else if (state==2){
    LembabSelada();
    lcd.setCursor(0,2);
    lcd.print("Lembab Bayam");
    lcd.setCursor(0,3);
    lcd.print("Tekan Tombol UP");
}
    lcd.setCursor(0,1);
    lcd.print("Lembab Sekarang=");
    lcd.print(humidity);
}

    delay(100);
    button_down=digitalRead(2);
    button_up=digitalRead(3);
    sel=digitalRead(4);
    metu=digitalRead(12);
    if(button_down==0){
        if(temp>0)
        {
            temp=temp-1;
        }
    }
    if(button_up==0){
        if(temp<4)
        {
            temp=temp+1;
        }
    }
    if(sel==0){
        if(kolom<7){
            kolom=kolom+1;
            //baris=baris+1;

```

```

        baris=temp;
        temp=0;
    }
    if(state==3){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("    ");
        lihat=1;
    }
    if(state==4){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("    ");
        lihat=2;
    }
    if(state==5){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("    ");
        lihat=3;
    }
}
if(metu==0){
    if(kolom>1)
        kolom=kolom-1;
    baris=0;
    temp=0;
}
switch(lihat){
case 1:
    lcd.clear();
    lcd.setCursor(5,0);
    lcd.print("Set Point");
    lcd.setCursor(0,1);
    lcd.print("Suhu=");
    lcd.setCursor(6,1);
    lcd.print(suhuS1);
    lcd.setCursor(0,2);
    lcd.print("Lembab=");

```

```

    lcd.setCursor(8,2);
    lcd.print(lembabS1);
    lcd.setCursor(0,3);
    lcd.print("Selada");
    break;
    case 2:
    lcd.clear();
    lcd.setCursor(5,0);
    lcd.print("Set Point");
    lcd.setCursor(0,1);
    lcd.print("Suhu=");
    lcd.setCursor(6,1);
    lcd.print(suhuS1);
    lcd.setCursor(0,2);
    lcd.print("Lembab=");
    lcd.setCursor(8,2);
    lcd.print(lembabS1);
    lcd.setCursor(0,3);
    lcd.print("Pak Coy");
    break;
    case 3:
    lcd.clear();
    lcd.setCursor(5,0);
    lcd.print("Set Point");
    lcd.setCursor(0,1);
    lcd.print("Suhu=");
    lcd.setCursor(6,1);
    lcd.print(suhuS1);
    lcd.setCursor(0,2);
    lcd.print("Lembab=");
    lcd.setCursor(8,2);
    lcd.print(lembabS1);
    lcd.setCursor(0,3);
    lcd.print("Bayam");
}
bt();
if (f2==1 & f==true){
    analogWrite(mot, 255);
    digitalWrite(in, LOW);
}

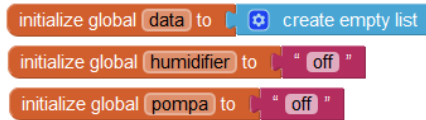
```

```

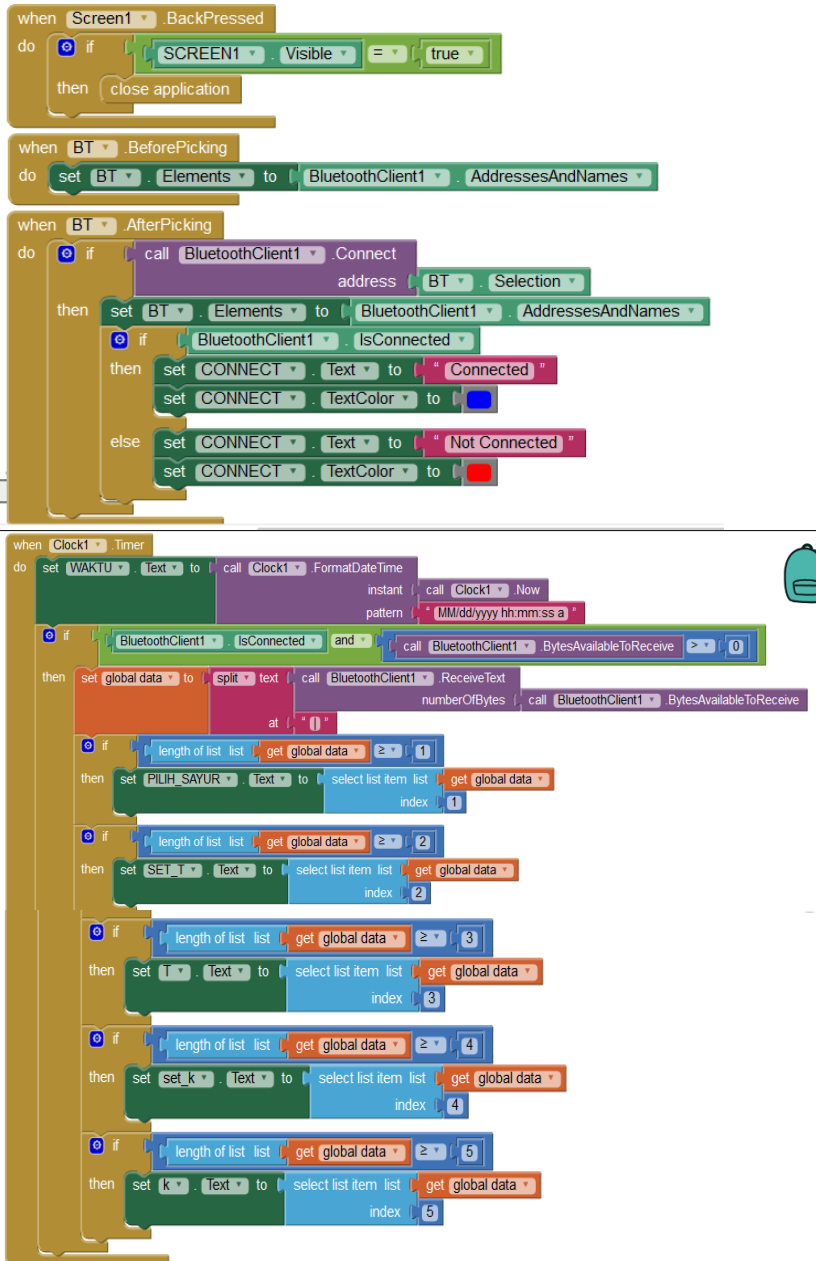
    digitalWrite(in1, HIGH);
    delay(1000);
    analogWrite(mot, 0);
    f=false;
    f1=true;
  }
  if (f2==2&&f1==true){
    analogWrite(mot, 255);
    digitalWrite(in, HIGH);
    digitalWrite(in1, LOW);
    delay(1000);
    analogWrite(mot, 0);
    f=true;
    f1=false;
  }
}

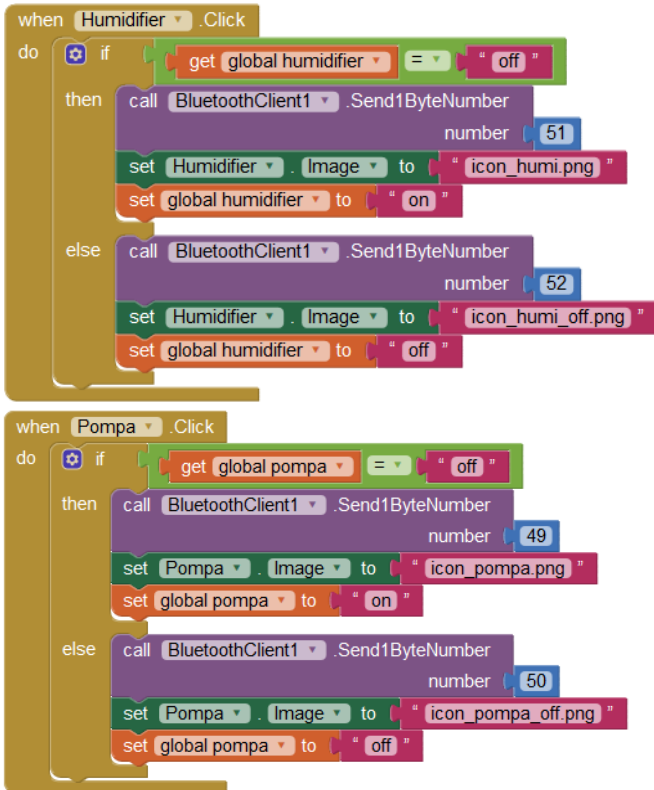
```

## A.2 Program MIT App Inventor









## LAMPIRAN B

### B.1 Pin Mapping Arduino Mega 2560

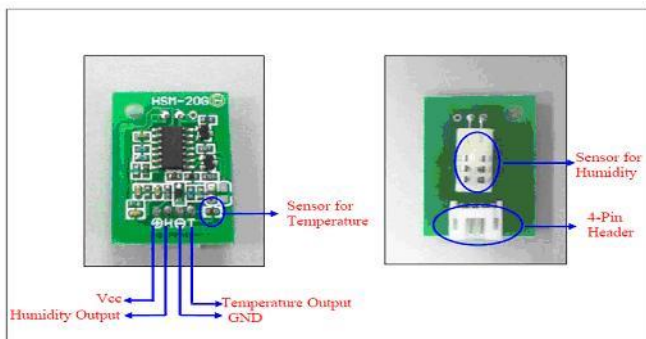
Pin	Nama Pin	Pemetaan Nama Pin
1.	PG5 (OCOB)	Digital Pin 4 (PWM)
2.	PE0 ( RXD0/PCINT8 )	Digital pin 0 (RX0)
3.	PE1 (TXD0)	Digital pin 1 (TX0)
4.	PE2(XCK0/AIN0)	-
5.	PE3 ( OC3A/AIN1 )	Digital pin 5 (PWM)
6.	PE4 ( OC3B/INT4 )	Digital pin 2 (PWM)
7.	PE5 ( OC3C/INT5 )	Digital pin 3 (PWM)
8.	PE6 ( T3/INT6 )	-
9.	PE7 (CLKO/ICP3/INT7)	-
10.	VCC	VCC
11.	GND	GND
12.	PH0 ( RXD2 )	Digital pin 17 (RX2)
13.	PH1 ( TXD2 )	Digital pin 16 (TX2)
14.	PH2 ( XCK2 )	-
15.	PH3 ( OC4A )	-
16.	PH4 ( OC4B )	Digital pin 7 (PWM)
17.	PH5 ( OC4C )	Digital pin 8 (PWM)
18.	PH6 ( OC2B )	Digital pin 9 (PWM)
19.	PB0 ( SS/PCINT0 )	Digital pin 53 (SS)
20.	PB1 ( SCK/PCINT1 )	Digital pin 52 (SCK)
21.	PB2 ( MOSI/PCINT2 )	Digital pin 51 (MOSI)
22.	PB3 ( MISO/PCINT3 )	Digital pin 50 (MISO)
23.	PB4 ( OC2A/PCINT4 )	Digital pin 10 (PWM)
24.	PB5 ( OC1A/PCINT5 )	Digital pin 11 (PWM)
25.	PB6 ( OC1B/PCINT6 )	Digital pin 12 (PWM)
26.	PB7 ( OC0A/OC1C/PCINT7 )	Digital pin 13 (PWM)
27.	PH7 ( T4 )	-
28.	PG3 ( TOSC2 )	-
29.	PG4 ( TOSC1 )	-
30.	RESET	RESET

<b>Pin</b>	<b>Nama Pin</b>	<b>Pemetaan Nama Pin</b>
31.	VCC	VCC
32.	GND	GND
33.	XTAL2	XTAL2
34.	XTAL1	XTAL1
35.	PL0 ( ICP4 )	Digital pin 49
36.	PL1 ( ICP5 )	Digital pin 48
37.	PL2 ( T5 )	Digital pin 47
38.	PL3 ( OC5A )	Digital pin 46 (PWM)
39.	PL4 ( OC5B )	Digital pin 45 (PWM)
40.	PL5 ( OC5C )	Digital pin 44 (PWM)
41.	PL6	Digital pin 43
42.	PL7	Digital pin 42
43.	PD0 ( SCL/INT0 )	Digital pin 21 (SCL)
44.	PD1 ( SDA/INT1 )	Digital pin 20 (SDA)
45.	PD2 ( RXDI/INT2 )	Digital pin 19 (RX1)
46.	PD3 ( TXDI/INT3 )	Digital pin 18 (TX1)
47.	PD4 ( ICP1 )	-
48.	PD5 ( XCK1 )	-
49.	PD6 ( T1 )	-
50.	PD7 ( T0 )	Digital pin 38
51.	PG0 ( WR )	Digital pin 41
52.	PG1 ( RD )	Digital pin 40
53.	PC0 ( A8 )	Digital pin 37
54.	PC1 ( A9 )	Digital pin 36
55.	PC2 ( A10 )	Digital pin 35
56.	PC3 ( A11 )	Digital pin 34
57.	PC4 ( A12 )	Digital pin 33
58.	PC5 ( A13 )	Digital pin 32

<b>Pin</b>	<b>Nama Pin</b>	<b>Pemetaan Nama Pin</b>
59.	PC6 ( A14 )	Digital pin 31
60.	PC7 ( A15 )	Digital pin 30
61.	VCC	VCC
62.	GND	GND
63.	PJ0 ( RXD3/PCINT9 )	Digital pin 15 (RX3)
64.	PJ1 ( TXD3/PCINT10 )	Digital pin 14 (TX3)
65.	PJ2 ( XCK3/PCINT11 )	-
66.	PJ3 ( PCINT12 )	-
67.	PJ4 ( PCINT13 )	-
68.	PJ5 ( PCINT14 )	-
69.	PJ6 ( PCINT 15 )	-
70.	PG2 ( ALE )	Digital pin 39
71.	PA7 ( AD7 )	Digital pin 29
72.	PA6 ( AD6 )	Digital pin 28
73.	PA5 ( AD5 )	Digital pin 27
74.	PA4 ( AD4 )	Digital pin 26
75.	PA3 ( AD3 )	Digital pin 25
76.	PA2 ( AD2 )	Digital pin 24
77.	PA1 ( AD1 )	Digital pin 23
78.	PA0 ( AD0 )	Digital pin 22
79.	PJ7	-
80.	VCC	VCC
81.	GND	GND
82.	PK7 ( ADC15/PCINT23 )	Analog pin 15
83.	PK6 ( ADC14/PCINT22 )	Analog pin 14
84.	PK5 ( ADC13/PCINT21 )	Analog pin 13
85.	PK4 ( ADC12/PCINT20 )	Analog pin 12

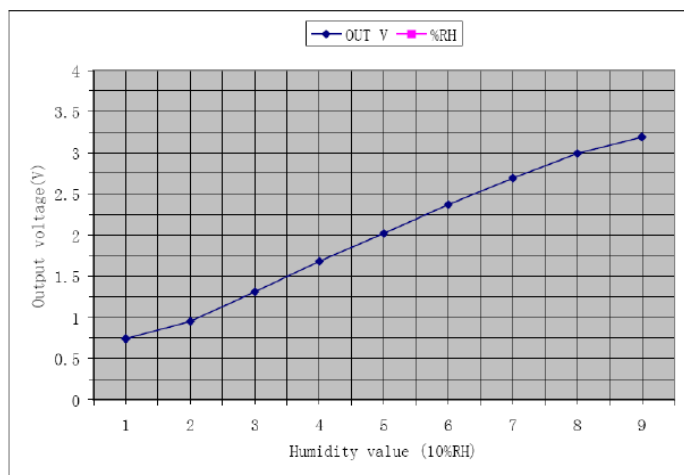
No	Nama Pin	Pemetaan Nama Pin
86.	PK3 ( ADC11/PCINT19 )	Analog pin 11
87.	PK2 ( ADC10/PCINT18 )	Analog pin 10
88.	PK1 ( ADC9/PCINT17 )	Analog pin 9
89.	PF7 ( ADC7 )	Analog pin 8
90.	PF7 ( ADC7 )	Analog pin 7
91.	PF6 ( ADC6 )	Analog pin 6
92.	PF5 ( ADC5/TMS )	Analog pin 5
93.	PF4 ( ADC4/TMK )	Analog pin 4
94.	PF3 ( ADC3 )	Analog pin 3
95.	PF2 ( ADC2 )	Analog pin 2
96.	PF1 ( ADC1 )	Analog pin 1
97.	PF0 ( ADC0 )	Analog pin 0
98.	AREF	Analog Referensi
99.	GND	GND
100.	AVCC	VCC

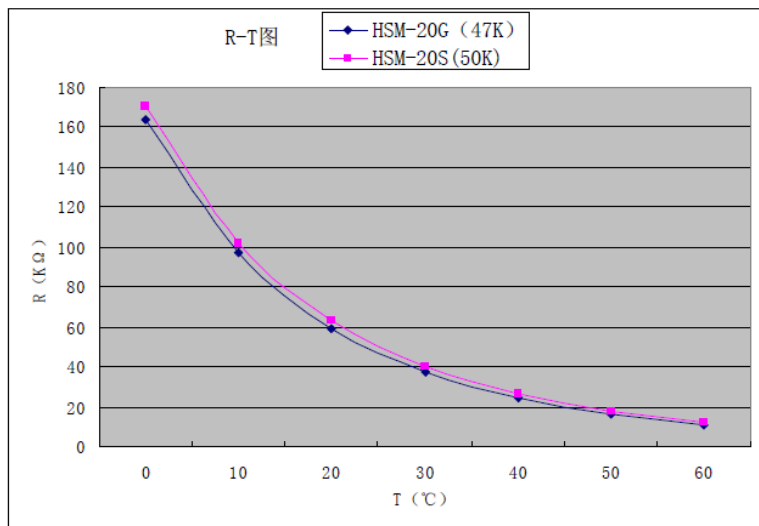
## B.2 HSM 20-G



Characteristics		HSM-20G
Input voltage range		DC 5.0±0.2V
Output voltage range		DC 1.0—3.0 V
Measurement Accuracy		±5% RH
Operating Current (Maximum)		2mA
Storage RH Range		0 to 99% RH
Operating RH Range		20 to 95% (100% RH intermittent)
Transient Condensation		< 3%RH
Temperature Range	Storage	-20℃ to 70℃
	Operating	0℃ to 50℃
Hysteresis (RH @ 25℃)		MAX 2%RH
Long Term Stability(typical drift per year)		±1.5%
Linearity		Linearity
Time Response(63% step change)		1 min
Dimensions(L*W)		34mm*22mm

#### 4. Typical Response of HSM-20G at 25 °C





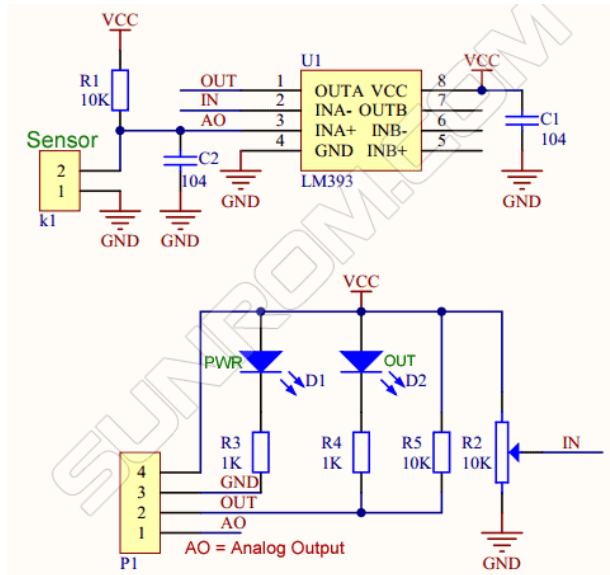
### B.3 Modul Light Dependent Resistor



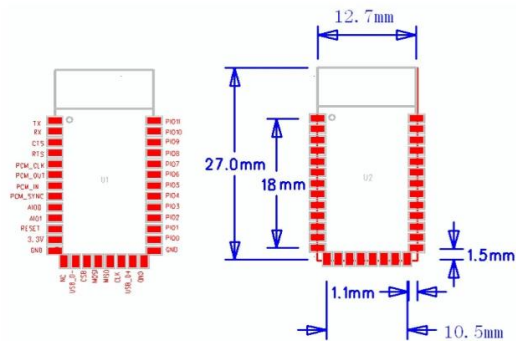
#### Pin details

- VCC = 3.3V to 5V DC
- GND = Ground
- DO = Digital Output
- AO = Analog Output

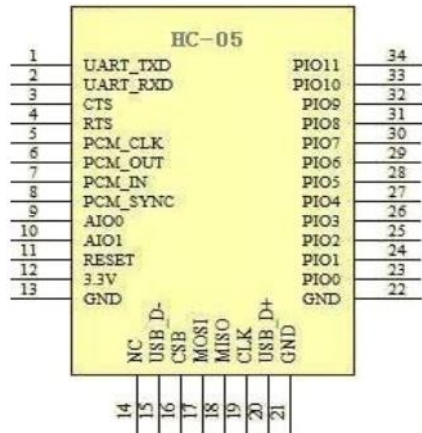




## B.2 Modul Bluetooth HC-05



### B.3 Konfigurasi Pin Modul *Bluetooth* HC-05



PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
3.3 VCC	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V	
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	
PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	
PIO7	30	Bi-Directional	Programmable input/output line	
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	

## B.4 Perintah AT Command

### Detailed description of Command

(AT command is case- sensitive, should end up with terminator ("enter" or "\r\n").)

#### 1. Test

Command	Response	Parameter
AT	OK	None

#### 2. Reset

Command	Response	Parameter
AT+RESET	OK	None

#### 3. Get the soft version

Command	Response	Parameter
AT+VERSION?	+VERSION: <Param> OK	Param: Version number

#### Example :

at+version?\r\n

+VERSION:2.0-20100601

OK

#### 4. Restore default status

Command	Response	Parameter
AT+ORGL	OK	None

The parameter of default status:

- ①. Device type: 0
- ②. Inquire code: 0x009e8b33
- ③. Module work mode: Slave Mode
- ④. Connection mode: Connect to the Bluetooth device specified
- ⑤. Serial parameter: Baud rate: 38400 bits/s; Stop bit: 1 bit; Parity bit: None.
- ⑥. Passkey: "1234"
- ⑦. Device name: "H-C-2010-06-01"

#### 5. Get module Bluetooth address

Command	Response	Parameter
AT+ADDR?	+ADDR: <Param> OK	Param: Bluetooth address

Bluetooth address will show as this way: NAP: UAP: LAP(Hexadecimal)

#### Example:

Module Bluetooth address: 12: 34: 56: ab: cd: ef

at+addr?\r\n

+ADDR:1234:56:abcdef

OK

#### 6. Set/ inquire device's name

Command	Response	Parameter
AT+NAME=<Param>	OK	
AT+NAME?	1. +NAME:<Param> OK----success 2. FAIL----failure	Param: Bluetooth device name Default: "HC-05"

#### Example:

AT+NAME=HC-05\r\n ---set the module device name: "HC-05"

OK

AT+NAME="HC-05"\r\n ---set the module device name: "HC-05"

OK

at+name=Beijin\r\n ---set the module device name: "Beijin"

OK

at+name="Beijin"\r\n ---set module device name : "Beijin"

OK

at+name?\r\n

+NAME: Bei jin

OK

#### 7. Get the remote Bluetooth device's name

Command	Response	Parameter
AT+RNAME?<Param1>	1. +NAME:<Param2> OK----success 2. FAIL----failure	Param1: Remote Bluetooth device address Param2: Remote Bluetooth device address

Bluetooth address will show as this way: NAP:UAP:LAP (Hexadecimal)

#### Example:

Bluetooth device address: 00:02:72: od: 22 : 24; device name: Bluetooth

at+name? 0002,72,od2224\r\n

+RNAME:Bluetooth

OK

#### 8. Set/ inquire module role

Command	Response	Parameter
AT+ROLE=<Param>	OK	Param: 0---- Slave role 1---- Master role 2---- Slave-Loop role Default: 0
AT+ROLE?	+ROLE:<Param> OK	

Role introduction:

Slave (slave role)----Passive connection;

Slave-Loop----Passive connection, receive the remote Bluetooth master device data and send it back to the master device;

Master (master role)----Inquire the near SPP Bluetooth slave device, build connection with it positively, and build up the transparent data transmission between master and slave device.

#### 9. Set/inquire device type

Command	Response	Parameter
AT+CLASS=<Param>	OK	Param: device type
AT+ CLASS?	1. + CLASS:<Param> OK----success 2. FAIL----failure	Bluetooth device type is a 32-bit parameter indicates the device type and what type can be supported. Default: 0 More information is provided at the appendix 1(device type introduction).

For inquiring the custom Bluetooth device from around Bluetooth devices quickly and effectively, user can set the module to be non-standard Bluetooth device type, such as 0x1f1f (Hexadecimal).

#### 12. Set/Inquire- passkey

Command	Response	Parameter
AT+PSWD=<Param>	OK	Param: passkey Default: "1234"
AT+ PSWD?	+ PSWD : <Param> OK	

### 13.Set/ Inquire- serial parameter

Command	Response	Parameter
AT+UART=<Param>,<Param2>,<Param3>	OK	Param1: baud rate( bits/s) The value (Decimal) should be one of the following: 4800 9600 19200 38400 57600 115200 23400 460800 921600 1382400 Param2: stop bit: 0----1 bit 1----2 bits Param3: parity bit 0----Even parity 1----Odd parity 2----Even parity Default: 9600, 0, 0
AT+ UART?	+ UART=<Param>,<Param2>,<Param3> OK	

#### Example:

Set baud rate to be 115200, stop bit to be 2 bits, parity bit to be even parity.

AT+UART=115200,1,2,\r\n

OK

AT+UART?

+UART:115200,1,2

OK

**Example:**

Set baud rate to be 115200, stop bit to be 2 bits, parity bit to be even parity.

AT+UART=115200,1,2,\r\n

OK

AT+UART?

+UART:115200,1,2

OK

**14. Set/ Inquire - connection mode**

Command	Response	Parameter
AT+CMODE=<Param>	OK	Param:
AT+CMODE?	+CMODE:<Param> OK	0----connect the module to the specified Bluetooth address. (Bluetooth address can be specified by the binding command) 1----connect the module to any address (The specifying address has no effect for this mode.) 2----Slave-Loop Default connection mode: 0

**15. Set/Inquire - bind Bluetooth address**

Bluetooth address will show as this way: NAP: UAP:LAP(Hexadecimal)



## LAMPIRAN C

### C.1 Dokumentasi



## C.2 Penyemaian Benih Selada Hari ke-4 (Senin, 10 Juli 2017)



b. Tabel Pengujian Pukul 09.00-09.15

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
28°C	58%	Nyala	2,59	Mati
28°C	60%	Nyala	2,51	Mati
28°C	60%	Nyala	2,22	Nyala
28°C	61%	Nyala	2,07	Nyala
27°C	61%	Nyala	1,56	Nyala

c. Tabel Pengujian Pukul 12.30-12.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
27°C	61%	Nyala	2,57	Mati
28°C	61%	Nyala	2,52	Mati
28°C	62%	Nyala	2,34	Nyala
29°C	61%	Nyala	2,22	Nyala
28°C	61%	Nyala	1,82	Nyala

d. Tabel Pengujian Pukul 16.30-16.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
27°C	60%	Nyala	2,67	Mati
27°C	61%	Nyala	2,58	Mati
27°C	61%	Nyala	2,52	Mati
27°C	61%	Nyala	1,72	Nyala
26°C	61%	Nyala	1,22	Nyala

### C.3. Penyemaian Benih Selada Hari ke-4 (Selasa, 11 Juli 2017)

a. Gambar Kondisi Benih Selada pada *Starter Kit*



b. Tabel Pengujian Pukul 09.00-09.15

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
27°C	57%	Nyala	2,73	Mati
27°C	58%	Nyala	2,6	Mati
27°C	58%	Nyala	2,51	Mati
27°C	59%	Nyala	2,35	Nyala
27°C	59%	Nyala	2,12	Nyala

c. Tabel Pengujian Pukul 12.30-12.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
28°C	55%	Nyala	2,67	Mati

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
28°C	56%	Nyala	2,52	Mati
28°C	56%	Nyala	2,43	Nyala
28°C	57%	Nyala	2,33	Nyala
28°C	57%	Nyala	2,21	Nyala

d. Tabel Pengujian Pukul 16.30-16.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
28°C	59%	Nyala	2,73	Mati
28°C	58%	Nyala	2,70	Mati
28°C	59%	Nyala	2,62	Mati
28°C	59%	Nyala	2,43	Nyala
28°C	59%	Nyala	2,37	Nyala

#### C.4 Penyemaian Benih Selada Hari ke-5 (Rabu, 12 Juli 2017)

a. Gambar Kondisi Benih Selada pada *Starter Kit*



b. Tabel Pengujian Pukul 09.00-09.15

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
28°C	60%	Nyala	2,63	Mati
28°C	60%	Nyala	2,57	Mati

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
29°C	60%	Nyala	2,55	Mati
28°C	60%	Nyala	2,21	Nyala
28°C	60%	Nyala	2,17	Nyala

c. Tabel Pengujian Pukul 12.30-12.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
30°C	60%	Nyala	2,75	Mati
29°C	60%	Nyala	2,67	Mati
29°C	60%	Nyala	2,55	Mati
29°C	60%	Nyala	2,36	Nyala
29°C	60%	Nyala	2,01	Nyala

d. Tabel Pengujian Pukul 16.30-16.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
28°C	59%	Nyala	2,71	Mati
28°C	59%	Nyala	2,65	Mati
28°C	60%	Nyala	2,51	Mati
28°C	59%	Nyala	2,26	Nyala
28°C	60%	Nyala	2,11	Nyala

## C.5 Penyemaian Benih Selada Hari ke-5 (Kamis, 13 Juli 2017)

a. Gambar Kondisi Benih Selada pada *Starter Kit*



b. Tabel Pengujian Pukul 09.00-09.15

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
28°C	59%	Nyala	2,63	Mati
28°C	59%	Nyala	2,62	Mati
28°C	59%	Nyala	2,57	Mati
28°C	60%	Nyala	2,42	Nyala
28°C	60%	Nyala	2,23	Nyala

c. Tabel Pengujian Pukul 12.30-12.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
29°C	56%	Nyala	2,72	Mati
29°C	56%	Nyala	2,51	Mati
28°C	57%	Nyala	2,43	Nyala
28°C	57%	Nyala	2,32	Nyala
28°C	57%	Nyala	2,10	Nyala

d. Tabel Pengujian Pukul 16.30-16.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
28°C	55%	Nyala	2,75	Mati
28°C	56%	Nyala	2,50	Mati
27°C	56%	Nyala	2,4	Nyala
27°C	56%	Nyala	2,31	Nyala
27°C	56%	Nyala	2,25	Nyala

### C.6 Penyemaian Benih Selada Hari ke-6 (Sabtu, 15 Juli 2017)

a. Gambar Kondisi Benih Selada pada *Starter Kit*



c. Tabel Pengujian Pukul 09.00-09.15

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
27°C	59%	Nyala	2,69	Mati
27°C	59%	Nyala	2,54	Mati
27°C	59%	Nyala	2,44	Nyala
27°C	59%	Nyala	2,37	Nyala
27°C	59%	Nyala	2,23	Nyala

d. Tabel Pengujian Pukul 12.30-12.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
29°C	57%	Nyala	2,78	Mati
29°C	58%	Nyala	2,62	Mati
29°C	58%	Nyala	2,55	Mati
28°C	59%	Nyala	2,43	Nyala
28°C	58%	Nyala	2,07	Nyala

e. Tabel Pengujian Pukul 16.30-16.45

Tempera- tur	Kelemba- ban	<i>Humidi- fier</i>	Ketinggian Air	Pompa
27°C	59%	Nyala	2,76	Mati
28°C	59%	Nyala	2,61	Mati
27°C	60%	Nyala	2,56	Mati
27°C	60%	Nyala	2,33	Nyala
27°C	60%	Nyala	2,2	Nyala



## RIWAYAT HIDUP PENULIS



Nama : Moh. Ilham Aziz W  
TTL : Nganjuk, 27 Juli 1996  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat : Jalan Irian, Ds.Bodor,  
Kec Pace, Kab.  
Nganjuk  
Telp/HP : 085649480515  
E-mail : *cakazizilham@gmail.com*

### RIWAYAT PENDIDIKAN:

1. 2002-2008 : SD Negeri 1 Bodor
2. 2008-2011 : SMP Negeri 1 Pace
3. 2011-2014 : SMA Negeri 1 Sukomoro
4. 2014-2017 : Program Studi Komputer Kontrol,  
Departemen Teknik Elektro Otomasi,  
Fakultas Vokasi, Institut Teknologi  
Sepuluh Nopember, Surabaya

### PENGALAMAN KERJA

1. Kerja Praktek di PT PLN APD JATIM (Persero) Surabaya

### PENGALAMAN ORGANISASI

1. PSDM HIMAD3TEKTRO 2015
2. Melukis Harapan Bidang Pendidikan 2014-2016
3. Argabayu-Paguyuban Mahasiswa Nganjuk-Surabaya 2014-2015
4. Asisten Laboratorium Otomasi dan Komputer D3 Teknik Elektro 2017

-----Halaman ini sengaja dikosongkan-----

## RIWAYAT HIDUP PENULIS



Nama : Safitri Febrianti  
TTL : Madiun, 17 Pebruari  
1996  
Jenis Kelamin : Perempuan  
Agama : Islam  
Alamat : Jalan Empusupo RT  
26 RW 09 Josenan,  
Kota Madiun  
Telp/HP : 082267111840  
E-mail : *safitriefbrianti29*  
*@gmail.com*

### RIWAYAT PENDIDIKAN:

1. 2002-2008 : MIN Demangan Kota Madiun
2. 2008-2011 : SMP Negeri 2 Madiun
3. 2011-2014 : SMA Negeri 1 Madiun
4. 2014-2017 : Program Studi Komputer Kontrol,  
Departemen Teknik Elektro Otomasi,  
Fakultas Vokasi, Institut Teknologi  
Sepuluh Nopember, Surabaya

### PENGALAMAN KERJA

1. Kerja Praktek di PT PLN (Persero) APB Jawa Timur, Sepanjang, Sidoarjo

### PENGALAMAN ORGANISASI

1. Departemen Dalam Negeri HIMAD3TEKTRO 2015-2016
2. Departemen Dalam Negeri HIMAD3TEKTRO 2016-2017
3. Asisten Laboratorium Elektronika Dasar D3 Teknik Elektro 2017

-----Halaman ini sengaja dikosongkan-----